Mathematical and Computational Applications

MDPI

Article

Reduced-Order Modelling and Homogenisation in Magneto-Mechanics: A Numerical Comparison of Established Hyper-Reduction Methods

Benjamin Brands *, Denis Davydov, Julia Mergheim and Paul Steinmann

Chair of Applied Mechanics, Friedrich-Alexander Universität Erlangen-Nürnberg, Egerlandstraße 5, 91058 Erlangen, Germany; denis.davydov@fau.de (D.D.); julia.mergheim@fau.de (J.M.); paul.steinmann@fau.de (P.S.)

* Correspondence: benjamin.brands@fau.de; Tel.: +49-(0)9131-85-64406

Received: 10 December 2018; Accepted: 29 January 2019; Published: 1 February 2019

Abstract: The simulation of complex engineering structures built from magneto-rheological elastomers is a computationally challenging task. Using the FE^2 method, which is based on computational homogenisation, leads to the repetitive solution of micro-scale FE problems, causing excessive computational effort. In this paper, the micro-scale FE problems are replaced by POD reduced models of comparable accuracy. As these models do not deliver the required reductions in computational effort, they are combined with hyper-reduction methods like the Discrete Empirical Interpolation Method (DEIM), Gappy POD, Gauss–Newton Approximated Tensors (GNAT), Empirical Cubature (EC) and Reduced Integration Domain (RID). The goal of this work is the comparison of the aforementioned hyper-reduction techniques focusing on accuracy and robustness. For the application in the FE^2 framework, EC and RID are favourable due to their robustness, whereas Gappy POD rendered both the most accurate and efficient reduced models. The well-known DEIM is discarded for this application as it suffers from serious robustness deficiencies.

Keywords: model order reduction; POD; DEIM; gappy POD; GNAT; ECSW; empirical cubature; hyper-reduction; reduced integration domain; computational homogenisation

1. Introduction

The ongoing development of so-called smart materials over the last decades has given rise to the quest for numerical models which enable predictive, fast and accurate simulations of engineering structures. For smart materials, the desired constitutive behaviour is frequently architectured by tailoring the microstructure of said material, e.g. fibre-reinforced composites, auxetic materials, metal foams and many more. An established approach to model such structures is denoted multiscale modelling for which commonly two scales, the micro- and macro-scale, are introduced. The geometric complexities and advanced boundary conditions of engineering structures are modelled on the macro-scale, whereas the microstructure is represented on the micro-scale. One way to consistently couple micro- and macro-scale is the so-called FE^2 method [1]. FE^2 is a multi-level finite element method that derives the constitutive response in every quadrature point of the macro-scale from an FE simulation incorporating the microstructure using the framework of computational homogenisation [2].

Even though the everlasting increase in computational resources following Moore's law has enabled scientists to solve FE problems with 10^{13} DoFs [3], the numerical cost of FE^2 simulations is still prohibitive for most realistic problems. The idea of replacing the micro-scale FE simulation by a less expensive model has brought together the fields of multiscale and reduced-order modelling.

In the last several years, several viable models targeted at reducing the multiscale FE simulation have been developed. In this contribution, we focus on projection-based models using a reduced

basis, but there are also alternatives like the Nonuniform Transformation Field Analysis [4,5] and Proper Generalized Decomposition [6,7]. In projection-based reduced models, the reduced basis is a set of few functions with global support that is constructed to approximate the solution manifold of the problem in question. Projecting the governing equations onto the reduced basis yields a considerable reduction in the number of unknowns compared to using the locally supported FE basis functions. The most commonly used methods to construct the reduced basis are Proper Orthogonal Decomposition (POD) [8–11] and the Reduced Basis Method [12–15]. Both rely on solutions of the parametrised partial differential equation (pPDE), for POD, the pPDE is solved for a set of given parameters, whereas the Reduced Basis Method employs a greedy algorithm equipped with an *a posteriori* error estimator to determine the parameters adaptively. As there are hardly any efficient and reliable error estimators for coupled nonlinear multi-physic problems, POD is the method of choice in this work. In [16], a POD reduced basis was used for the first time in multiscale analysis of nonlinear elasticity at finite strains, namely by reducing the micro-scale model. This was extended in [17] by introducing the computation of a consistent tangent operator based on the reduced model.

For problems with nonlinearities or non-affine parameter dependence, the sole application of a reduced basis does not render the desired computational savings as the nonlinearity or non-affine parameter dependence has to be evaluated for the original model and subsequently projected onto the reduced basis. A widely used method accelerating the computation of the nonlinearity is the (Discrete) Empirical Interpolation Method (D)EIM [18,19]. DEIM approximates the nonlinearity by a linear combination of collateral basis functions. The coefficients are computed using interpolation based on values of the nonlinearity sampled at a relatively small number of points. In order to improve the approximation, interpolation is replaced by linear regression for Gappy POD [20]. In [21], Petrov–Galerkin projection is used to increase the stability of reduced models. Together with Gappy POD and possibly differing approximations of the reduced system matrix, this is referred to as GNAT (Gauss–Newton Approximated Tensors). As DEIM, Gappy POD and GNAT use collateral basis functions to approximate the nonlinearity, they are classified as collateral basis methods. Both POD and DEIM have been applied previously to various mechanical problems: a simplified beam model for multiscale modelling at small strains including damage [22], strain-softening viscoplasticity at small strains [23] and structural mechanics using a variant of DEIM based on the unassembled nonlinearity [24]. A collateral basis for the stresses instead of the nonlinearity was used in [25] for homogenisation of elasto-plastic materials at small strains, together with Gappy POD and a tailored method to determine the locations at which the stresses are evaluated in the reduced model. A detailed survey of DEIM, Gappy POD and GNAT for homogenisation of hyper-elastic materials at finite strains that focus on accuracy and robustness was performed in [26].

Cubature methods are another possibility of reducing the cost of computing the nonlinearity. In this sense, a problem specific quadrature rule replaces the quadrature used to integrate the weak form, e.g. Gaussian quadrature. This empirically determined quadrature uses only a subset of the support points or elements of the original FE model and computes the weights accordingly. This idea was put forward in [27] and later introduced to the field of computational homogenisation as Energy-Conserving Sampling and Weighting (ECSW) [28]. A possibility to reduce the cost of constructing the cubature was introduced in [29] together with the term Empirical Cubature (EC). The accuracy and efficiency of EC was compared to a variant of DEIM/Gappy POD used in [25] for homogenisation of elasto-viscoplastic materials in small strains [30].

The Hyper-Reduction method [31] makes use of a reduced integration domain (RID) to speed up the computation of the nonlinearity. It defines test functions with support confined to the RID, which in combination with trial functions obtained by POD results in a Petrov–Galerkin projection. The expression hyper-reduction was coined in [31] but is now used as a term encompassing all methods aiming at accelerating the computation of nonlinearities in the field of model reduction. Therefore, we will refer to the Hyper-Reduction method [31] as RID to avoid any notational confusion. The RID was used for the simulation of elasto-plasticity [32], the simulation of nonlinear thermal and

mechanical problems involving internal variables [33] and the lifetime assessment of elasto-plastic structures [34]. An algorithmic comparison with DEIM for the nonlinear heat equation was carried out in [35]. Similarly, the Missing Point Estimation (MPE) method [36] computes the Galerkin projection in a small subset of the computational domain to accelerate the assembly of the reduced problem. An investigation of the MPE method is beyond the scope of this article and accordingly we refer the interested reader to [37], where a detailed comparison of MPE, DEIM and Gappy POD was performed for a predator–prey model.

In this contribution, we will show the first application of reduced-order modelling for computational homogenisation in magneto-mechanics at finite strains. We will focus on reducing the problem at the micro-scale, using POD to compute the reduced basis and applying following hyper-reduction methods: DEIM, Gappy POD, GNAT, EC and RID. Through various numerical studies, a thorough comparison between the techniques with emphasis on accuracy and robustness will be drawn.

2. Homogenisation in Magneto-Mechanics

The simulation of engineering structures requires evaluations of a material law at the engineering/macro-scale (≈mm–m). For magneto-rheological elastomers (MREs), the constitutive behaviour on the macro-scale is determined by the underlying microstructure (≈nm–µm). Usually, MREs are composite materials consisting of an elastomeric matrix with embedded magneto-active particles [38] which induce changes in stiffness or deformations due to applied magnetic fields. Due to the scale separation, a resolution of the microstructure in the discretisation of the macrostructure is computationally not feasible. The tools of computational homogenisation offer an expedient to the issue as the constitutive behaviour for any point on the macro-scale is computed from the solution of a boundary value problem (BVP) representative for the microstructure. The material composition of the microstructure is described by an RVE (Representative Volume Element) for which the constitutive behaviour of the constituents is prescribed. In the context of magneto-mechanics, the macroscopic deformation gradient $\overline{\boldsymbol{F}}$ and the magnetic field $\overline{\mathbb{H}}$ are the input variables for the microstructural BVP [39,40]. In the remainder, we use an over-bar to denote macro variables $\overline{(\bullet)}$.

As is common in homogenisation, the micro displacement and scalar magnetic potential are additively split into two parts, the macroscopic fields and the fluctuations:

$$\boldsymbol{u} = \overline{\boldsymbol{F}} \cdot \boldsymbol{X} + \tilde{\boldsymbol{u}} \qquad \mathbb{y} = \overline{\mathbb{H}} \cdot \boldsymbol{X} + \tilde{\mathbb{y}}. \tag{1}$$

The macroscopic fields depend linearly on the macroscopic deformation gradient $\overline{\boldsymbol{F}}$, the macroscopic magnetic field $\overline{\mathbb{H}}$ and the position vector $\boldsymbol{X}$.

We use linear boundary conditions to fulfill the Hill–Mandel condition. Using the fluctuations $\tilde{\boldsymbol{u}}$ and $\tilde{\mathbb{y}}$ as primary variables allows us to transform the linear into homogeneous boundary conditions. The RVE occupies the domain $\mathcal{B}_0 \subset \mathbb{R}^{\mathrm{d}}$ with its boundary $\partial\mathcal{B}_0$, where d denotes the space dimension. The energy density $\Psi(\boldsymbol{F}, \mathbb{H})$ is expressed in terms of the deformation gradient $\boldsymbol{F}$ and the magnetic field $\mathbb{H}$ and used to define the constitutive relations for the Piola stress $\boldsymbol{P}$ and the magnetic induction $\mathbb{B}$. The balance of linear momentum and Gauss's law for magnetism (also known as conservation of magnetic flux) complete the strong form of magneto-mechanics on the micro-scale [41]:

$$\begin{aligned}
&\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{u} = \overline{\boldsymbol{F}} + \nabla_{\boldsymbol{X}} \tilde{\boldsymbol{u}} && \mathbb{H} = \nabla_{\boldsymbol{X}} \mathbb{y} = \overline{\mathbb{H}} + \nabla_{\boldsymbol{X}} \tilde{\mathbb{y}} && \text{in } \mathcal{B}_0, \\
&\boldsymbol{P} = \frac{\partial \Psi(\boldsymbol{F}, \mathbb{H})}{\partial \boldsymbol{F}} && \mathbb{B} = -\frac{\partial \Psi(\boldsymbol{F}, \mathbb{H})}{\partial \mathbb{H}} && \text{in } \mathcal{B}_0, \\
&\operatorname{Div} \boldsymbol{P} = \boldsymbol{0} && \operatorname{Div} \mathbb{B} = 0 && \text{in } \mathcal{B}_0, \\
&\tilde{\boldsymbol{u}} = \boldsymbol{0} && \tilde{\mathbb{y}} = 0 && \text{on } \partial\mathcal{B}_0.
\end{aligned} \tag{2}$$

For the sake of an FE solution, the weak form

$$\begin{aligned} &\int_{\mathcal{B}_0} \nabla_{\boldsymbol{X}} \delta\tilde{\boldsymbol{u}} : \boldsymbol{P}\, \mathrm{d}V = 0 \quad \forall \delta\tilde{\boldsymbol{u}} \in \left\{ \delta\tilde{\boldsymbol{u}} \in H^1(\mathcal{B}_0) : \delta\tilde{\boldsymbol{u}} = \boldsymbol{0} \text{ on } \partial\mathcal{B}_0 \right\} \\ \text{and } &\int_{\mathcal{B}_0} \nabla_{\boldsymbol{X}} \delta\tilde{\mathrm{y}} \cdot \mathbb{B}\, \mathrm{d}V = 0 \quad \forall \delta\tilde{\mathrm{y}} \in \left\{ \delta\tilde{\mathrm{y}} \in H^1(\mathcal{B}_0) : \delta\tilde{\mathrm{y}} = 0 \text{ on } \partial\mathcal{B}_0 \right\} \end{aligned} \tag{3}$$

is derived using the test functions $\delta\tilde{\boldsymbol{u}}$ and $\delta\tilde{\mathrm{y}}$.

For the spatial discretisation, the standard Bubnov–Galerkin FEM is used. The continuum body is approximated by a mesh $\mathcal{B}_0 \approx \mathcal{T} = \bigcup_{e=1}^{\mathrm{M}} \Omega_e$ with $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$ and $i, j \in [1, \ldots, \mathrm{M}]$, where M denotes the number of elements. The displacement and potential fields in any finite element Ω_e are approximated by the piecewise continuous vector-valued polynomials $\boldsymbol{N}_i^{\boldsymbol{u}}(\boldsymbol{X})$ and scalar polynomials $N_i^{\mathrm{y}}(\boldsymbol{X})$, respectively:

$$\begin{aligned} \tilde{\boldsymbol{u}}|_{\Omega_e} &\coloneqq \sum_{i=1}^{\mathrm{d}_e^{\boldsymbol{u}}} \tilde{u}_i \boldsymbol{N}_i^{\boldsymbol{u}}(\boldsymbol{X}), & \delta\tilde{\boldsymbol{u}}|_{\Omega_e} &\coloneqq \sum_{i=1}^{\mathrm{d}_e^{\boldsymbol{u}}} \delta\tilde{u}_i \boldsymbol{N}_i^{\boldsymbol{u}}(\boldsymbol{X}), \\ \tilde{\mathrm{y}}|_{\Omega_e} &\coloneqq \sum_{i=1}^{\mathrm{d}_e^{\mathrm{y}}} \tilde{\mathrm{y}}_i N_i^{\mathrm{y}}(\boldsymbol{X}), & \delta\tilde{\mathrm{y}}|_{\Omega_e} &\coloneqq \sum_{i=1}^{\mathrm{d}_e^{\mathrm{y}}} \delta\tilde{\mathrm{y}}_i N_i^{\mathrm{y}}(\boldsymbol{X}). \end{aligned} \tag{4}$$

The scalars $\mathrm{d}_e^{\boldsymbol{u}}$ and $\mathrm{d}_e^{\mathrm{y}}$ are the numbers of mechanical and magnetic DoFs in the element Ω_e. Using these approximations results in the discrete weak form

$$\hat{\boldsymbol{R}} = \begin{bmatrix} \hat{\boldsymbol{R}}^{\boldsymbol{u}} \in \mathbb{R}^{\mathrm{N}_{\boldsymbol{u}}} \\ \hat{\boldsymbol{R}}^{\mathrm{y}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{y}}} \end{bmatrix} = \mathop{\mathscr{A}}_{e=1}^{\mathrm{M}} \begin{bmatrix} {}^{e}\hat{\boldsymbol{R}}^{\boldsymbol{u}} \in \mathbb{R}^{\mathrm{d}_e^{\boldsymbol{u}}} \\ {}^{e}\hat{\boldsymbol{R}}^{\mathrm{y}} \in \mathbb{R}^{\mathrm{d}_e^{\mathrm{y}}} \end{bmatrix} = \boldsymbol{0} \quad \text{with} \quad \begin{bmatrix} {}^{e}\hat{\boldsymbol{R}}^{\boldsymbol{u}}[i] = \int_{\Omega_e} \nabla_{\boldsymbol{X}} \boldsymbol{N}_i^{\boldsymbol{u}} : \boldsymbol{P}\, \mathrm{d}V \\ {}^{e}\hat{\boldsymbol{R}}^{\mathrm{y}}[i] = \int_{\Omega_e} \nabla_{\boldsymbol{X}} N_i^{\mathrm{y}} \cdot \mathbb{B}\, \mathrm{d}V \end{bmatrix} \text{ for } \Omega_e \in \mathcal{T}. \tag{5}$$

The sub-/superscripts $(\bullet)^{\boldsymbol{u}}/(\bullet)_{\boldsymbol{u}}$ and $(\bullet)^{\mathrm{y}}/(\bullet)_{\mathrm{y}}$ encode whether a variable is associated with the mechanical or magnetic component and are used throughout the remainder of the paper. The notation $\hat{(\bullet)}$ is consistently used to differentiate between a continuous field and its discrete FE counterpart, e.g. $\tilde{\boldsymbol{u}}$ is the displacement fluctuation and $\hat{\tilde{\boldsymbol{u}}}$ is the vector containing the nodal coefficients for the FE discretisation. To refer to single elements of any vector/first-order tensor $\boldsymbol{X}$ and of any matrix/second-order tensor $\boldsymbol{Y}$, the notation $\boldsymbol{X}[i]$ and $\boldsymbol{Y}[i,j]$ are used. The operator $\mathop{\mathscr{A}}_{e=1}^{\mathrm{M}}$ represents the assembly of the element contributions and the scalars $\mathrm{N}_{\boldsymbol{u}}$, N_{y} and $\mathrm{N} = \mathrm{N}_{\boldsymbol{u}} + \mathrm{N}_{\mathrm{y}}$ are the numbers of DoFs employed in the FE discretisation.

The numerical solution of the system of nonlinear equations (5) using the iterative Newton–Raphson scheme requires its linearisation

$$\boldsymbol{K}_k \Delta\hat{\tilde{\boldsymbol{y}}}_k = -\hat{\boldsymbol{R}}_k \;\text{ with } \hat{\tilde{\boldsymbol{y}}}_{k+1} = \hat{\tilde{\boldsymbol{y}}}_k + \Delta\hat{\tilde{\boldsymbol{y}}}_k \;\text{ and } \hat{\tilde{\boldsymbol{y}}}_k = \begin{bmatrix} \hat{\tilde{\boldsymbol{u}}}_k \\ \hat{\tilde{\mathrm{y}}}_k \end{bmatrix}, \tag{6}$$

introducing the iteration count k.

For the sake of notational clarity, the dependences of $\hat{\boldsymbol{R}}_k\left(\hat{\tilde{\boldsymbol{y}}}_k; \overline{\boldsymbol{F}}, \overline{\mathbb{H}}\right)$ and $\boldsymbol{K}_k\left(\hat{\tilde{\boldsymbol{y}}}_k; \overline{\boldsymbol{F}}, \overline{\mathbb{H}}\right)$ are dropped. The tangent stiffness matrix $\boldsymbol{K}$ is given as

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}^{\boldsymbol{uu}} \in \mathbb{R}^{\mathrm{N}_{\boldsymbol{u}} \times \mathrm{N}_{\boldsymbol{u}}} & \boldsymbol{K}^{\boldsymbol{u}\mathrm{y}} \in \mathbb{R}^{\mathrm{N}_{\boldsymbol{u}} \times \mathrm{N}_{\mathrm{y}}} \\ \boldsymbol{K}^{\mathrm{y}\boldsymbol{u}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{y}} \times \mathrm{N}_{\boldsymbol{u}}} & \boldsymbol{K}^{\mathrm{y}\mathrm{y}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{y}} \times \mathrm{N}_{\mathrm{y}}} \end{bmatrix} = \mathop{\mathrm{M}}_{e=1}^{\mathscr{A}} \begin{bmatrix} {}^{e}\boldsymbol{K}^{\boldsymbol{uu}} \in \mathbb{R}^{\mathrm{d}_e^{\boldsymbol{u}} \times \mathrm{d}_e^{\boldsymbol{u}}} & {}^{e}\boldsymbol{K}^{\boldsymbol{u}\mathrm{y}} \in \mathbb{R}^{\mathrm{d}_e^{\boldsymbol{u}} \times \mathrm{d}_e^{\mathrm{y}}} \\ {}^{e}\boldsymbol{K}^{\mathrm{y}\boldsymbol{u}} \in \mathbb{R}^{\mathrm{d}_e^{\mathrm{y}} \times \mathrm{d}_e^{\boldsymbol{u}}} & {}^{e}\boldsymbol{K}^{\mathrm{y}\mathrm{y}} \in \mathbb{R}^{\mathrm{d}_e^{\mathrm{y}} \times \mathrm{d}_e^{\mathrm{y}}} \end{bmatrix}$$

$$\text{with} \quad \begin{aligned} {}^{e}\boldsymbol{K}^{\boldsymbol{uu}}[i,j] &= \int_{\Omega_e} \nabla_{\boldsymbol{X}} \boldsymbol{N}_i^{\boldsymbol{u}} : \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}} : \nabla_{\boldsymbol{X}} \boldsymbol{N}_j^{\boldsymbol{u}} \, \mathrm{d}V \\ {}^{e}\boldsymbol{K}^{\boldsymbol{u}\mathrm{y}}[i,j] &= \int_{\Omega_e} \nabla_{\boldsymbol{X}} \boldsymbol{N}_i^{\boldsymbol{u}} : \frac{\partial \boldsymbol{P}}{\partial \mathbb{H}} \cdot \nabla_{\boldsymbol{X}} N_j^{\mathrm{y}} \, \mathrm{d}V \\ {}^{e}\boldsymbol{K}^{\mathrm{y}\boldsymbol{u}}[i,j] &= \int_{\Omega_e} \nabla_{\boldsymbol{X}} N_i^{\mathrm{y}} \cdot \frac{\partial \mathbb{B}}{\partial \boldsymbol{F}} : \nabla_{\boldsymbol{X}} \boldsymbol{N}_j^{\boldsymbol{u}} \, \mathrm{d}V \\ {}^{e}\boldsymbol{K}^{\mathrm{y}\mathrm{y}}[i,j] &= \int_{\Omega_e} \nabla_{\boldsymbol{X}} N_i^{\mathrm{y}} \cdot \frac{\partial \mathbb{B}}{\partial \mathbb{H}} \cdot \nabla_{\boldsymbol{X}} N_j^{\mathrm{y}} \, \mathrm{d}V \end{aligned} \quad \text{for } \Omega_e \in \mathcal{T}. \tag{7}$$

Once the solution of (5) is obtained, the output quantities $\overline{\boldsymbol{P}}$ and $\overline{\mathbb{B}}$ are computed using

$$\overline{\boldsymbol{P}} = \frac{1}{V} \int_{\mathcal{B}_0} \boldsymbol{P} \, \mathrm{d}V \quad \text{and} \quad \overline{\mathbb{B}} = \frac{1}{V} \int_{\mathcal{B}_0} \mathbb{B} \, \mathrm{d}V, \tag{8}$$

where the volume of the RVE is denoted by V.

3. Reduced-Order Modelling

3.1. Reduced Basis

Instead of using a large number N of compact trial functions, projection-based ROMs are built upon a small number n of global functions spanning the space in which the solution manifold of the pPDE resides. Consequently, the unknown fluctuation fields $\tilde{\boldsymbol{u}}$ and $\tilde{\mathrm{y}}$ are expressed as linear combinations of the global trial functions with the reduced coefficients $\tilde{u}_i^r$ and $\tilde{\mathrm{y}}_i^r$:

$$\tilde{\boldsymbol{u}} = \sum_{i=1}^{\mathrm{n}_{\boldsymbol{u}}} \boldsymbol{\varphi}_i^{\boldsymbol{u}} \tilde{u}_i^r \quad \text{and} \quad \tilde{\mathrm{y}} = \sum_{i=1}^{\mathrm{n}_{\mathrm{y}}} \varphi_i^{\mathrm{y}} \tilde{\mathrm{y}}_i^r. \tag{9}$$

The reducibility of the problem, namely the conditions $\mathrm{n}_{\boldsymbol{u}} \ll \mathrm{N}_{\boldsymbol{u}}$ and $\mathrm{n}_{\mathrm{y}} \ll \mathrm{N}_{\mathrm{y}}$, is accepted implicitly but has to be confirmed by numerical studies. To avoid scaling issues due to differently chosen units, separate reduced bases are used for the mechanical and the magnetic fluctuation fields. The numbers $\mathrm{n}_{\boldsymbol{u}}$, n_{y} and $\mathrm{n} = \mathrm{n}_{\boldsymbol{u}} + \mathrm{n}_{\mathrm{y}}$ are the numbers of reduced basis functions to be taken into account.

A common method to compute the reduced basis for a pPDE is POD [9,42]. In order to do so, we define the parameter domain of the microscopic problem

$$\begin{aligned} \mathcal{P} = &\left(\overline{\boldsymbol{F}}^{\min}[1,1], \overline{\boldsymbol{F}}^{\max}[1,1] \right) \times \cdots \times \left(\overline{\boldsymbol{F}}^{\min}[\mathrm{d},\mathrm{d}], \overline{\boldsymbol{F}}^{\max}[\mathrm{d},\mathrm{d}] \right) \\ &\times \left(\overline{\mathbb{H}}^{\min}[1], \overline{\mathbb{H}}^{\max}[1] \right) \times \cdots \times \left(\overline{\mathbb{H}}^{\min}[\mathrm{d}], \overline{\mathbb{H}}^{\max}[\mathrm{d}] \right) \subset \mathbb{R}^{\mathrm{d}^2+\mathrm{d}} \end{aligned} \tag{10}$$

with reasonably chosen limits for the components of the macroscopic loading parameters. Each element $\mathbf{p}_i = \left(\overline{\boldsymbol{F}}_i, \overline{\mathbb{H}}_i \right) \in \mathcal{P}$ comprises an instance of the macroscopic deformation gradient and magnetic field. The parameter domain $\mathcal{P}$ is sampled using n_s parameters gathered in the set

$$\mathcal{S} = \left\{ \mathbf{p}_1, \ldots, \mathbf{p}_{\mathrm{n}_s} \right\} \subset \mathcal{P} \tag{11}$$

and the full-order model (FOM) (5) is solved for all elements in $\mathcal{S}$. The solutions are collected in the snapshot matrices

$$\boldsymbol{S_u} = \left[\hat{\tilde{\boldsymbol{u}}}(\mathbf{p}_1), \ldots, \hat{\tilde{\boldsymbol{u}}}(\mathbf{p}_{\mathrm{n_s}})\right] \in \mathbb{R}^{\mathrm{N_u} \times \mathrm{n_s}} \quad \text{and} \quad \boldsymbol{S}_{\mathrm{y}} = \left[\hat{\tilde{\mathrm{y}}}(\mathbf{p}_1), \ldots, \hat{\tilde{\mathrm{y}}}(\mathbf{p}_{\mathrm{n_s}})\right] \in \mathbb{R}^{\mathrm{N_y} \times \mathrm{n_s}} \tag{12}$$

and the subsequent application of

$$\mathrm{POD}\left(\boldsymbol{S_u}\right) \to \boldsymbol{B_u} = \left[\hat{\boldsymbol{\varphi}}^{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{\varphi}}^{\boldsymbol{u}}_{\mathrm{n_u}}\right] \in \mathbb{R}^{\mathrm{N_u} \times \mathrm{n_u}} \quad \text{and} \quad \mathrm{POD}\left(\boldsymbol{S}_{\mathrm{y}}\right) \to \boldsymbol{B}_{\mathrm{y}} = \left[\hat{\boldsymbol{\varphi}}^{\mathrm{y}}_1, \ldots, \hat{\boldsymbol{\varphi}}^{\mathrm{y}}_{\mathrm{n_y}}\right] \in \mathbb{R}^{\mathrm{N_y} \times \mathrm{n_y}} \tag{13}$$

gives the discrete reduced bases contained in the matrices $\boldsymbol{B_u}$ and $\boldsymbol{B}_{\mathrm{y}}$. For details on POD, we refer to [9,43].

3.2. Galerkin ROM

In the Galerkin reduced model, the same ansatz (9) as for the solution is used for the test functions

$$\delta\tilde{\boldsymbol{u}} = \sum_{i=1}^{\mathrm{n_u}} \delta\tilde{u}^r_i \boldsymbol{\varphi}^{\boldsymbol{u}}_i \quad \text{and} \quad \delta\tilde{\mathrm{y}} = \sum_{i=1}^{\mathrm{n_y}} \delta\tilde{\mathrm{y}}^r_i \varphi^{\mathrm{y}}_i . \tag{14}$$

Inserting (9) and (14) into (5) results in the weak form of the Galerkin reduced model

$$\int_{\mathcal{B}_0} \nabla_{\boldsymbol{X}} \boldsymbol{\varphi}^{\boldsymbol{u}}_i : \boldsymbol{P}\, \mathrm{d}V = 0 \quad \forall i = 1, \ldots, \mathrm{n_u} \quad \text{and} \quad \int_{\mathcal{B}_0} \nabla_{\boldsymbol{X}} \varphi^{\mathrm{y}}_i \cdot \mathbb{B}\, \mathrm{d}V = 0 \quad \forall i = 1, \ldots, \mathrm{n_y}, \tag{15}$$

where the dependences $\boldsymbol{P}\left(\tilde{\boldsymbol{u}}, \tilde{\mathrm{y}}; \overline{\boldsymbol{F}}, \overline{\mathbb{H}}\right)$ and $\mathbb{B}\left(\tilde{\boldsymbol{u}}, \tilde{\mathrm{y}}; \overline{\boldsymbol{F}}, \overline{\mathbb{H}}\right)$ are dropped for notational brevity.

Analogously, the discrete weak (16) form and its linearisation (17) are derived:

$$\boldsymbol{B}^{\top} \hat{\boldsymbol{R}} = \boldsymbol{0} \quad \text{with } \boldsymbol{B} := \begin{bmatrix} \boldsymbol{B_u} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_{\mathrm{y}} \end{bmatrix} \in \mathbb{R}^{\mathrm{N} \times \mathrm{n}}, \tag{16}$$

$$\boldsymbol{B}^{\top} \boldsymbol{K}_k \boldsymbol{B} \Delta\tilde{\boldsymbol{y}}^r_k = -\boldsymbol{B}^{\top} \hat{\boldsymbol{R}}_k \quad \text{with } \tilde{\boldsymbol{y}}^r_{k+1} = \tilde{\boldsymbol{y}}^r_k + \Delta\tilde{\boldsymbol{y}}^r_k \text{ and } \tilde{\boldsymbol{y}}^r_k = \begin{bmatrix} \tilde{\boldsymbol{u}}^r_k \in \mathbb{R}^{\mathrm{n_u}} \\ \tilde{\mathrm{y}}^r_k \in \mathbb{R}^{\mathrm{n_y}} \end{bmatrix}. \tag{17}$$

Even though the size of the system of linear equations (17) is significantly smaller than in Equation (6) and hence the cost of the linear solver reduces from $\mathcal{O}(\mathrm{N}^2)$ to $\mathcal{O}(\mathrm{n}^3)$, the speed-up is only marginal as the assembly of (17) depends on the original problem size. The cost for evaluating the constitutive law for every quadrature point is roughly $\mathcal{O}(\mathrm{Nn} + \mathrm{n}^{\mathrm{el}}_{\mathrm{qp}}\mathrm{M})$, where $\mathrm{n}^{\mathrm{el}}_{\mathrm{qp}}$ is the number of quadrature points per element. The computational complexities of assembling and projecting $\boldsymbol{B}^{\top}\hat{\boldsymbol{R}}_k$ and $\boldsymbol{B}^{\top}\boldsymbol{K}_k\boldsymbol{B}$ are proportional to $\mathcal{O}(\mathrm{nN})$ and $\mathcal{O}(\mathrm{n}^2\mathrm{N} + \mathrm{nN})$, respectively. Therefore, the application of hyper-reduction methods is imperative.

4. Hyper-Reduction

4.1. Discrete Empirical Interpolation Method

The Discrete Empirical Interpolation Method [19] is the standard hyper-reduction method for non-affine or nonlinear pPDEs, for some problems even equipped with *a posteriori* and *a priori* error estimators [44,45]. The first step is to approximate the discrete residuum

$$\begin{aligned} &\hat{\boldsymbol{R}} = \begin{bmatrix} \hat{\boldsymbol{R}}^{\boldsymbol{u}} \\ \hat{\boldsymbol{R}}^{\mathrm{y}} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{R}^{\boldsymbol{u}}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_{\boldsymbol{R}^{\mathrm{y}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{r_u} \\ \boldsymbol{r}_{\mathrm{y}} \end{bmatrix} =: \boldsymbol{H}_{\boldsymbol{R}} \boldsymbol{r}, \\ &\text{with } \boldsymbol{H}_{\boldsymbol{R}^{\boldsymbol{u}}} = \left[\hat{\boldsymbol{\phi}}^{\boldsymbol{R}^{\boldsymbol{u}}}_1, \ldots, \hat{\boldsymbol{\phi}}^{\boldsymbol{R}^{\boldsymbol{u}}}_{\mathrm{r_u}}\right] \in \mathbb{R}^{\mathrm{N_u} \times \mathrm{r_u}}, \; \boldsymbol{H}_{\boldsymbol{R}^{\mathrm{y}}} = \left[\hat{\boldsymbol{\phi}}^{\boldsymbol{R}^{\mathrm{y}}}_1, \ldots, \hat{\boldsymbol{\phi}}^{\boldsymbol{R}^{\mathrm{y}}}_{\mathrm{r_y}}\right] \in \mathbb{R}^{\mathrm{N_y} \times \mathrm{r_y}}, \\ &\boldsymbol{r_u} \in \mathbb{R}^{\mathrm{r_u}}, \; \boldsymbol{r}_{\mathrm{y}} \in \mathbb{R}^{\mathrm{r_y}}, \; \boldsymbol{H}_{\boldsymbol{R}} \in \mathbb{R}^{\mathrm{N} \times \mathrm{r}} \text{ and } \boldsymbol{r} \in \mathbb{R}^{\mathrm{r}} \end{aligned} \tag{18}$$

by a linear combination of collateral basis vectors contained in $\boldsymbol{H_R}$ with $\boldsymbol{r}$ being the vector of coefficients. Due to a different number range, it is advisable to approximate the mechanical and magnetic residua by two separate collateral bases $\boldsymbol{H}_{\boldsymbol{R}^u}$ and $\boldsymbol{H}_{\boldsymbol{R}^y}$. The collateral basis is computed based on snapshots of the residuum. For that purpose, (16) is solved for every parameter $\mathbf{p}_i \in \mathcal{S}$ and the residua

$$\boldsymbol{T}_i^u = \left[\hat{\boldsymbol{R}}_1^u\left(\mathbf{p}_i\right), \ldots, \hat{\boldsymbol{R}}_{k_u}^u\left(\mathbf{p}_i\right)\right] \quad \text{and} \quad \boldsymbol{T}_i^y = \left[\hat{\boldsymbol{R}}_1^y\left(\mathbf{p}_i\right), \ldots, \hat{\boldsymbol{R}}_{k_y}^y\left(\mathbf{p}_i\right)\right] \tag{19}$$

are collected in the course of the Newton–Raphson process to build the matrices

$$\boldsymbol{S}_{\boldsymbol{R}^u} = \left[\boldsymbol{T}_1^u, \ldots, \boldsymbol{T}_{\mathrm{n_s}}^u\right] \quad \text{and} \quad \boldsymbol{S}_{\boldsymbol{R}^y} = \left[\boldsymbol{T}_1^y, \ldots, \boldsymbol{T}_{\mathrm{n_s}}^y\right]. \tag{20}$$

As $\hat{\boldsymbol{R}}_j^u\left(\mathbf{p}_i\right)$ and $\hat{\boldsymbol{R}}_j^y\left(\mathbf{p}_i\right)$ converge to the null vector during the iterative solution of (16), only residua fulfilling $\left\|\hat{\boldsymbol{R}}_j^u(\mathbf{p}_i)\right\| / \left\|\hat{\boldsymbol{R}}_1^u(\mathbf{p}_i)\right\| > \text{tol}$ and $\left\|\hat{\boldsymbol{R}}_j^y(\mathbf{p}_i)\right\| / \left\|\hat{\boldsymbol{R}}_1^y(\mathbf{p}_i)\right\| > \text{tol}$ are taken into account. The subsequent application of POD $\left(\boldsymbol{S}_{\boldsymbol{R}^u}\right) \rightarrow \boldsymbol{H}_{\boldsymbol{R}^u}$ and POD $\left(\boldsymbol{S}_{\boldsymbol{R}^y}\right) \rightarrow \boldsymbol{H}_{\boldsymbol{R}^y}$ gives the collateral bases.

The coefficients in (18) are determined using interpolation

$$\begin{aligned} \mathbf{P}_{\boldsymbol{R}^u}^\top \hat{\boldsymbol{R}}^u &= \mathbf{P}_{\boldsymbol{R}^u}^\top \boldsymbol{H}_{\boldsymbol{R}^u} \boldsymbol{r}_u \qquad \text{with } \mathbf{P}_{\boldsymbol{R}^u} = \left[\boldsymbol{e}_{\rho_1^u}, \ldots, \boldsymbol{e}_{\rho_{\mathrm{r}_u}^u}\right] \in \mathbb{N}^{\mathrm{N}_u \times \mathrm{r}_u}, \\ \mathbf{P}_{\boldsymbol{R}^y}^\top \hat{\boldsymbol{R}}^y &= \mathbf{P}_{\boldsymbol{R}^y}^\top \boldsymbol{H}_{\boldsymbol{R}^y} \boldsymbol{r}_y \qquad \text{with } \mathbf{P}_{\boldsymbol{R}^y} = \left[\boldsymbol{e}_{\rho_1^y}, \ldots, \boldsymbol{e}_{\rho_{\mathrm{r}_y}^y}\right] \in \mathbb{N}^{\mathrm{N}_y \times \mathrm{r}_y}, \end{aligned} \tag{21}$$

meaning the approximation has to be equal to the residuum at the interpolation indices. The matrices $\mathbf{P}_{\boldsymbol{R}^u}$ and $\mathbf{P}_{\boldsymbol{R}^y}$ are sampling matrices, where $\boldsymbol{e}_{\rho_i^u}$ for $i = 1, \ldots, \mathrm{r}_u$ and $\boldsymbol{e}_{\rho_j^y}$ for $j = 1, \ldots, \mathrm{r}_y$ are unit vectors with only one non-zero component in the ρ_i^u-th and ρ_j^y-th entry.

Application of the DEIM Algorithm 1, provided in Appendix A, to the collateral bases $\left\{\hat{\boldsymbol{\phi}}_i^{R^u}\right\}_{i=1}^{\mathrm{r}_u}$ and $\left\{\hat{\boldsymbol{\phi}}_i^{R^y}\right\}_{i=1}^{\mathrm{r}_y}$ returns the interpolation indices and guarantees the matrix products $\left[\mathbf{P}_{\boldsymbol{R}^u}^\top \boldsymbol{H}_{\boldsymbol{R}^u}\right]$ and $\left[\mathbf{P}_{\boldsymbol{R}^u}^\top \boldsymbol{H}_{\boldsymbol{R}^u}\right]$ to be non-singular. Consequently, the interpolation coefficients are calculated by

$$\boldsymbol{r}_u = \left(\mathbf{P}_{\boldsymbol{R}^u}^\top \boldsymbol{H}_{\boldsymbol{R}^u}\right)^{-1} \mathbf{P}_{\boldsymbol{R}^u}^\top \hat{\boldsymbol{R}}^u \quad \text{and} \quad \boldsymbol{r}_y = \left(\mathbf{P}_{\boldsymbol{R}^y}^\top \boldsymbol{H}_{\boldsymbol{R}^y}\right)^{-1} \mathbf{P}_{\boldsymbol{R}^y}^\top \hat{\boldsymbol{R}}^y. \tag{22}$$

Introducing (18) and (21) into (16) gives the hyper-reduced weak form

$$\boldsymbol{R}_k^r := \boldsymbol{B}^\top \boldsymbol{H}_{\boldsymbol{R}} \left(\mathbf{P}_{\boldsymbol{R}}^\top \boldsymbol{H}_{\boldsymbol{R}}\right)^{-1} \mathbf{P}_{\boldsymbol{R}}^\top \hat{\boldsymbol{R}}_k = \mathbf{0} \quad \text{with } \mathbf{P}_{\boldsymbol{R}} := \begin{bmatrix} \mathbf{P}_{\boldsymbol{R}^u} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\boldsymbol{R}^y} \end{bmatrix} \in \mathbb{N}^{\mathrm{N} \times \mathrm{r}} \tag{23}$$

and its linearisation becomes

$$\underbrace{\boldsymbol{B}^\top \boldsymbol{H}_{\boldsymbol{R}} \left(\mathbf{P}_{\boldsymbol{R}}^\top \boldsymbol{H}_{\boldsymbol{R}}\right)^{-1}}_{\text{precomputed: } \mathbb{R}^{\mathrm{n} \times \mathrm{r}}} \mathbf{P}_{\boldsymbol{R}}^\top \boldsymbol{K}_k \boldsymbol{B} \Delta \tilde{\boldsymbol{y}}_k^r = - \underbrace{\boldsymbol{B}^\top \boldsymbol{H}_{\boldsymbol{R}} \left(\mathbf{P}_{\boldsymbol{R}}^\top \boldsymbol{H}_{\boldsymbol{R}}\right)^{-1}}_{\text{precomputed: } \mathbb{R}^{\mathrm{n} \times \mathrm{r}}} \mathbf{P}_{\boldsymbol{R}}^\top \hat{\boldsymbol{R}}_k. \tag{24}$$

The cost for evaluating the constitutive law for every quadrature point in elements containing interpolation indices is reduced to approximately $\mathcal{O}(\mathrm{N_{eval}} \mathrm{n} + \mathrm{n_{qp}^{el}} \mathrm{m})$, where m is the number of elements containing DEIM indices and $\mathrm{N_{eval}}$ the number of DoFs associated with this elements. The computational complexities of computing the residuum and tangent stiffness matrix are proportional to $\mathcal{O}(\mathrm{rN_{eval}})$ and $\mathcal{O}(\mathrm{nrN_{eval}} + \mathrm{rN_{eval}})$, respectively. It is to be noted that an efficient computation of the stiffness matrix utilises the sparsity of the FE matrix. Consequently, the assembly and solution of (24) do not depend on the size of the FOM and should therefore result in the desired speed-ups.

4.2. Gappy POD

Instead of interpolation, Gappy POD uses linear regression to determine the collateral basis coefficients, meaning the residual is evaluated at more indices than coefficients. This is particularly beneficial for hyper-reduced models originating from FE models, as for the calculation of $\hat{\boldsymbol{R}}^{\boldsymbol{u}}$ or $\hat{\boldsymbol{R}}^{\mathrm{y}}$ at each evaluation index $\rho_i^{\boldsymbol{u}}$ or ρ_i^{y} the solution $(\tilde{\boldsymbol{u}}, \tilde{\mathrm{y}})$ and the respective constitutive components have to be computed for every finite element containing the index. Hence, it is more economical to use all DoFs attached to a node instead of possibly only one as for DEIM. The collateral basis coefficients are computed solving

$$\begin{aligned} \boldsymbol{r}_{\boldsymbol{u}} &= \underset{\boldsymbol{a}\in\mathbb{R}^{\mathrm{r}_{\boldsymbol{u}}}}{\arg\min} \left\| \mathbf{P}_{\boldsymbol{R}^{\boldsymbol{u}}}^{\top} \hat{\boldsymbol{R}}^{\boldsymbol{u}} - \mathbf{P}_{\boldsymbol{R}^{\boldsymbol{u}}}^{\top} \boldsymbol{H}_{\boldsymbol{R}^{\boldsymbol{u}}} \boldsymbol{a} \right\|_2^2 \quad \text{with } \mathbf{P}_{\boldsymbol{R}^{\boldsymbol{u}}} = \left[\boldsymbol{e}_{\rho_1^{\boldsymbol{u}}}, \ldots, \boldsymbol{e}_{\rho_{\mathrm{p}_{\boldsymbol{u}}\mathrm{d}}^{\boldsymbol{u}}} \right] \in \mathbb{N}^{\mathrm{N}_{\boldsymbol{u}} \times \mathrm{p}_{\boldsymbol{u}}\mathrm{d}} \\ \text{and } \boldsymbol{r}_{\mathrm{y}} &= \underset{\boldsymbol{a}\in\mathbb{R}^{\mathrm{r}_{\mathrm{y}}}}{\arg\min} \left\| \mathbf{P}_{\boldsymbol{R}^{\mathrm{y}}}^{\top} \hat{\boldsymbol{R}}^{\mathrm{y}} - \mathbf{P}_{\boldsymbol{R}^{\mathrm{y}}}^{\top} \boldsymbol{H}_{\boldsymbol{R}^{\mathrm{y}}} \boldsymbol{a} \right\|_2^2 \quad \text{with } \mathbf{P}_{\boldsymbol{R}^{\mathrm{y}}} = \left[\boldsymbol{e}_{\rho_1^{\mathrm{y}}}, \ldots, \boldsymbol{e}_{\rho_{\mathrm{p}_{\mathrm{y}}}^{\mathrm{y}}} \right] \in \mathbb{N}^{\mathrm{N}_{\mathrm{y}} \times \mathrm{p}_{\mathrm{y}}}. \end{aligned} \tag{25}$$

The integers $\mathrm{p}_{\boldsymbol{u}}$ and p_{y} are the numbers of FE nodes at which the residua $\hat{\boldsymbol{R}}^{\boldsymbol{u}}$ and $\hat{\boldsymbol{R}}^{\mathrm{y}}$ are computed and $\mathrm{N}_{\boldsymbol{u}}$ and N_{y} are the dimensions of the underlying FE model. For (25) to have unique solutions, $\mathrm{p}_{\boldsymbol{u}}\mathrm{d} \geq \mathrm{r}_{\boldsymbol{u}}$ and $\mathrm{p}_{\mathrm{y}} \geq \mathrm{r}_{\mathrm{y}}$ have to hold. The solutions of (25) are obtained by computing the pseudo-inverses $\left(\mathbf{P}_{\boldsymbol{R}^{\boldsymbol{u}}}^{\top} \boldsymbol{H}_{\boldsymbol{R}^{\boldsymbol{u}}}\right)^{+}$ and $\left(\mathbf{P}_{\boldsymbol{R}^{\mathrm{y}}}^{\top} \boldsymbol{H}_{\boldsymbol{R}^{\mathrm{y}}}\right)^{+}$ rendering the explicit expressions

$$\boldsymbol{r}_{\boldsymbol{u}} = \left(\mathbf{P}_{\boldsymbol{R}^{\boldsymbol{u}}}^{\top} \boldsymbol{H}_{\boldsymbol{R}^{\boldsymbol{u}}}\right)^{+} \mathbf{P}_{\boldsymbol{R}^{\boldsymbol{u}}}^{\top} \hat{\boldsymbol{R}}^{\boldsymbol{u}} \quad \text{and} \quad \boldsymbol{r}_{\mathrm{y}} = \left(\mathbf{P}_{\boldsymbol{R}^{\mathrm{y}}}^{\top} \boldsymbol{H}_{\boldsymbol{R}^{\mathrm{y}}}\right)^{+} \mathbf{P}_{\boldsymbol{R}^{\mathrm{y}}}^{\top} \hat{\boldsymbol{R}}^{\mathrm{y}} \tag{26}$$

for the collateral basis coefficients.

Inserting (26) into (16) gives the Gappy POD hyper-reduced weak form

$$\boldsymbol{R}_k^r := \boldsymbol{B}^{\top} \boldsymbol{H}_{\boldsymbol{R}} \left(\mathbf{P}_{\boldsymbol{R}}^{\top} \boldsymbol{H}_{\boldsymbol{R}}\right)^{+} \mathbf{P}_{\boldsymbol{R}}^{\top} \hat{\boldsymbol{R}}_k = \boldsymbol{0} \tag{27}$$

and the linearisation becomes

$$\underbrace{\boldsymbol{B}^{\top} \boldsymbol{H}_{\boldsymbol{R}} \left(\mathbf{P}_{\boldsymbol{R}}^{\top} \boldsymbol{H}_{\boldsymbol{R}}\right)^{+}}_{\text{precomputed: } \mathbb{R}^{\mathrm{n}\times\mathrm{p}}} \mathbf{P}_{\boldsymbol{R}}^{\top} \boldsymbol{K}_k \boldsymbol{B} \Delta\tilde{\boldsymbol{y}}_k^r = - \underbrace{\boldsymbol{B}^{\top} \boldsymbol{H}_{\boldsymbol{R}} \left(\mathbf{P}_{\boldsymbol{R}}^{\top} \boldsymbol{H}_{\boldsymbol{R}}\right)^{+}}_{\text{precomputed: } \mathbb{R}^{\mathrm{n}\times\mathrm{p}}} \mathbf{P}_{\boldsymbol{R}}^{\top} \hat{\boldsymbol{R}}_k, \tag{28}$$

introducing $\mathrm{p} = \mathrm{p}_{\boldsymbol{u}}\mathrm{d} + \mathrm{p}_{\mathrm{y}}$. In (28), the only difference to the DEIM hyper-reduced system (24) is the appearance of the pseudo-inverse $(\bullet)^{+}$ instead of the inverse $(\bullet)^{-1}$. The cost for evaluating the constitutive law is the same as for DEIM $\mathcal{O}(\mathrm{N}_{\mathrm{eval}}\mathrm{n} + \mathrm{n}_{\mathrm{qp}}^{\mathrm{el}}\mathrm{m})$. The computational complexities of computing the residuum and tangent stiffness matrix are proportional to $\mathcal{O}(\mathrm{pN}_{\mathrm{eval}})$ and $\mathcal{O}(\mathrm{npN}_{\mathrm{eval}} + \mathrm{pN}_{\mathrm{eval}})$.

To determine the FE nodes/indices, Algorithm 2, given in Appendix B, which is an advancement of the algorithm proposed in [21] for multi-physic problems, is applied. Algorithm 2 uses normalised maxima to cope with distinct domains and different units in multi-physic problems. Algorithm 2 is applied to $\left\{\hat{\boldsymbol{\phi}}_1^{\boldsymbol{R}^{\boldsymbol{u}}}, \ldots, \hat{\boldsymbol{\phi}}_{\mathrm{r}_{\boldsymbol{u}}}^{\boldsymbol{R}^{\boldsymbol{u}}}\right\}$ and $\left\{\hat{\boldsymbol{\phi}}_1^{\boldsymbol{R}^{\mathrm{y}}}, \ldots, \hat{\boldsymbol{\phi}}_{\mathrm{r}_{\mathrm{y}}}^{\boldsymbol{R}^{\mathrm{y}}}\right\}$ either separately or combined. In the latter case, the same FE nodes are used for the gappy reconstruction of the residua $\hat{\boldsymbol{R}}^{\boldsymbol{u}}$ and $\hat{\boldsymbol{R}}^{\mathrm{y}}$, resulting in more efficient reduced models.

4.3. GNAT

To improve the accuracy and stability [21] of reduced models, the GNAT hyper-reduced model is not based on Galerkin but on Petrov–Galerkin projection and therefore adopts different spaces for the test and trial functions.

For accuracy reasons, the state-dependent test functions $\boldsymbol{K}\left(\boldsymbol{B}\boldsymbol{y}^r\right)\boldsymbol{B}$ are chosen and the discrete weak form

$$\boldsymbol{B}^\top \boldsymbol{K}^\top \hat{\boldsymbol{R}} = \boldsymbol{0} \tag{29}$$

and its linearisation

$$\boldsymbol{B}^\top \boldsymbol{K}_k^\top \boldsymbol{K}_k \boldsymbol{B} \Delta\tilde{\boldsymbol{y}}^r = -\boldsymbol{B}^\top \hat{\boldsymbol{R}}_k \quad \text{with} \quad \sum_i \hat{\boldsymbol{R}}[i] \frac{\partial^2 \hat{\boldsymbol{R}}[i]}{\partial \hat{\boldsymbol{y}}^2} \approx \boldsymbol{0} \tag{30}$$

are obtained. Equation (30) is the normal equation for the associated least-squares problem

$$\Delta\tilde{\boldsymbol{y}}_k^r = \underset{\boldsymbol{a}\in\mathbb{R}^{\mathrm{n}}}{\arg\min} \left\| \boldsymbol{K}_k \boldsymbol{B} \boldsymbol{a} + \hat{\boldsymbol{R}}_k \right\|_2^2 \tag{31}$$

and therefore the solution of (29) is equivalent to solving the minimisation problem

$$\underset{\boldsymbol{a}\in\mathbb{R}^{\mathrm{n}}}{\text{minimise}} \left\| \hat{\boldsymbol{R}}\left(\boldsymbol{B}\boldsymbol{a}\right) \right\|_2 \tag{32}$$

with the Gauss–Newton method.

As the computation of (31) still depends on the original problem size, GNAT similarly to Gappy POD uses collateral bases to approximate the nonlinearities and linear regression to determine the coefficients:

$$\begin{aligned} &\hat{\boldsymbol{R}} = \boldsymbol{H}_{\boldsymbol{R}} \boldsymbol{r} \quad \text{and} \quad \boldsymbol{K}\boldsymbol{B} = \boldsymbol{H}_{\boldsymbol{K}} \boldsymbol{k}, \\ &\text{with } \boldsymbol{r} = \left(\mathbf{P}_{\boldsymbol{R}}^\top \boldsymbol{H}_{\boldsymbol{R}}\right)^+ \mathbf{P}_{\boldsymbol{R}}^\top \hat{\boldsymbol{R}} \quad \text{and} \quad \boldsymbol{k} = \left(\mathbf{P}_{\boldsymbol{K}}^\top \boldsymbol{H}_{\boldsymbol{K}}\right)^+ \mathbf{P}_{\boldsymbol{K}}^\top \left[\boldsymbol{K}\boldsymbol{B}\right]. \end{aligned} \tag{33}$$

Numerical experiments have shown that the choice $\boldsymbol{H}_{\boldsymbol{R}} = \boldsymbol{H}_{\boldsymbol{K}}$ and consequently $\mathbf{P}_{\boldsymbol{R}} = \mathbf{P}_{\boldsymbol{K}}$ renders reduced models of superior accuracy compared to models employing a separate basis for $\boldsymbol{H}_{\boldsymbol{K}}$. Putting (33) into (31) and multiplying from the left with $\boldsymbol{H}_{\boldsymbol{R}}^\top$ renders the least-squares problem to be solved in every Gauss–Newton iteration

$$\Delta\tilde{\boldsymbol{y}}_k^r = \underset{\boldsymbol{a}\in\mathbb{R}^{\mathrm{n}}}{\arg\min} \left\| \underbrace{\left(\mathbf{P}_{\boldsymbol{R}}^\top \boldsymbol{H}_{\boldsymbol{R}}\right)^+}_{\text{precomputed: } \mathbb{R}^{\mathrm{r}\times\mathrm{p}}} \mathbf{P}_{\boldsymbol{R}}^\top \left[\boldsymbol{K}_k \boldsymbol{B}\right] \boldsymbol{a} + \overbrace{\underbrace{\left(\mathbf{P}_{\boldsymbol{R}}^\top \boldsymbol{H}_{\boldsymbol{R}}\right)^+}_{\text{precomputed: } \mathbb{R}^{\mathrm{r}\times\mathrm{p}}} \mathbf{P}_{\boldsymbol{R}}^\top \hat{\boldsymbol{R}}_k}^{\boldsymbol{R}_k^r} \right\|_2^2 \tag{34}$$

and recalling $\mathrm{p} = \mathrm{p}_{\boldsymbol{u}}\mathrm{d} + \mathrm{p}_{\mathrm{y}}$.

The complexity of assembling and solving (34) is similar to (28). As the Gauss–Newton method does not converge quadratically like the classical Newton–Raphson scheme, more iterations are necessary to minimise (32).

The computation of the collateral basis is similar to DEIM except that the residua are gathered during the solution of (31). The matrix $\mathbf{P}_{\boldsymbol{R}}$ is determined using Algorithm 2 with the collateral bases as input.

4.4. Empirical Cubature

Cubature methods aim at reducing the cost of computing the nonlinearity in (15) by defining an empirical quadrature, which evaluates the integrand only in a limited number of quadrature points or elements. Instead of summing up all element contributions, the nonlinearities are computed only in the elements of the so-called reduced meshes $\mathcal{E}_{\boldsymbol{u}}$ and $\mathcal{E}_{\mathrm{y}}$ and multiplied by positive weights:

$$\begin{aligned}
\left[\boldsymbol{B}_{\boldsymbol{u}}^{\top}\hat{\boldsymbol{R}}^{\boldsymbol{u}}\right][i] &= \sum_{e=1}^{\mathrm{M}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_i^{\boldsymbol{u}}:\boldsymbol{P}\,\mathrm{d}V \approx \sum_{e\in\mathcal{E}_{\boldsymbol{u}}}\omega_e^{\boldsymbol{u}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_i^{\boldsymbol{u}}:\boldsymbol{P}\,\mathrm{d}V \quad \text{for } i=1,\ldots,\mathrm{n}_{\boldsymbol{u}} \\
\text{and} \quad \left[\boldsymbol{B}_{\mathbb{y}}^{\top}\hat{\boldsymbol{R}}^{\mathbb{y}}\right][i] &= \sum_{e=1}^{\mathrm{M}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\varphi_i^{\mathbb{y}}\cdot\mathbb{B}\,\mathrm{d}V \approx \sum_{e\in\mathcal{E}_{\mathbb{y}}}\omega_e^{\mathbb{y}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\varphi_i^{\mathbb{y}}\cdot\mathbb{B}\,\mathrm{d}V \quad \text{for } i=1,\ldots,\mathrm{n}_{\mathbb{y}}, \\
&\text{with } \mathcal{E}_{\boldsymbol{u}}=\left\{e\in\{1,\ldots,\mathrm{M}\}:\omega_e^{\boldsymbol{u}}>0\right\} \quad \text{and} \quad \mathcal{E}_{\mathbb{y}}=\left\{e\in\{1,\ldots,\mathrm{M}\}:\omega_e^{\mathbb{y}}>0\right\}.
\end{aligned} \tag{35}$$

In (35), each element e in $\mathcal{E}_{\boldsymbol{u}}$ or $\mathcal{E}_{\mathbb{y}}$ is equipped with a positive weight $\omega_e^{\boldsymbol{u}}$ or $\omega_e^{\mathbb{y}}$, whereas all the other elements are assigned weights $\omega_e^{\boldsymbol{u}}=\omega_e^{\mathbb{y}}=0$.

The approximation in (35) induces the errors

$$\begin{aligned}
{}^{\boldsymbol{u}}e_{ij} &= \sum_{e\in\mathcal{E}_{\boldsymbol{u}}}\omega_e^{\boldsymbol{u}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_i^{\boldsymbol{u}}:\boldsymbol{P}_j\,\mathrm{d}V-\sum_{e=1}^{\mathrm{M}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_i^{\boldsymbol{u}}:\boldsymbol{P}_j\,\mathrm{d}V \\
\text{and} \quad {}^{\mathbb{y}}e_{ij} &= \sum_{e\in\mathcal{E}_{\mathbb{y}}}\omega_e^{\mathbb{y}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\varphi_i^{\mathbb{y}}\cdot\mathbb{B}_j\,\mathrm{d}V-\sum_{e=1}^{\mathrm{M}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\varphi_i^{\mathbb{y}}\cdot\mathbb{B}_j\,\mathrm{d}V
\end{aligned} \tag{36}$$

for snapshots of the stress field $\left\{\boldsymbol{P}_j\right\}_{j=1}^{\mathrm{n_s}}$ and the magnetic induction $\left\{\mathbb{B}_j\right\}_{j=1}^{\mathrm{n_s}}$. The reduced meshes $\mathcal{E}_{\boldsymbol{u}}$ and $\mathcal{E}_{\mathbb{y}}$ equipped with the weights $\{\omega_e^{\boldsymbol{u}}\}_{e=1}^{\mathrm{m}_{\boldsymbol{u}}}$ and $\{\omega_e^{\mathbb{y}}\}_{e=1}^{\mathrm{m}_{\mathbb{y}}}$ are constructed by minimising the errors (36), with $\mathrm{m}_{\boldsymbol{u}}$ and $\mathrm{m}_{\mathbb{y}}$ being the number of elements in the reduced meshes. Different algorithms for the minimisation of (36) are discussed in [46].

Since this minimisation is numerically expensive, collateral bases [29]

$$\boldsymbol{P}=\sum_{j=1}^{\mathrm{n}_{\boldsymbol{P}}}c_j^{\boldsymbol{P}}\boldsymbol{\phi}_j^{\boldsymbol{P}} \quad \text{and} \quad \mathbb{B}=\sum_{j=1}^{\mathrm{n}_{\mathbb{B}}}c_j^{\mathbb{B}}\boldsymbol{\phi}_j^{\mathbb{B}} \tag{37}$$

for the stress and induction fields are introduced, where $\mathrm{n}_{\boldsymbol{P}}\ll\mathrm{n_s}$ and $\mathrm{n}_{\mathbb{B}}\ll\mathrm{n_s}$ should hold. For that reason, (15) is solved for the parameters in $\mathcal{S}$ (11) and the snapshots of the stress and induction fields are gathered in the matrices $\boldsymbol{S}_{\boldsymbol{P}}$ and $\boldsymbol{S}_{\mathbb{B}}$. A successive application of POD gives the collateral bases $\boldsymbol{H}_{\boldsymbol{P}}$ and $\boldsymbol{H}_{\mathbb{B}}$:

$$\begin{aligned}
&\mathrm{POD}\left(\boldsymbol{S}_{\boldsymbol{P}}\right)\rightarrow\boldsymbol{H}_{\boldsymbol{P}}=\left[\hat{\boldsymbol{\phi}}_1^{\boldsymbol{P}},\ldots,\hat{\boldsymbol{\phi}}_{\mathrm{n}_{\boldsymbol{P}}}^{\boldsymbol{P}}\right]\in\mathbb{R}^{\mathrm{n_{qp}d^2}\times\mathrm{n}_{\boldsymbol{P}}} \quad \text{with } \boldsymbol{S}_{\boldsymbol{P}}=\left[\hat{\boldsymbol{P}}(\mathbf{p}_1),\ldots,\hat{\boldsymbol{P}}(\mathbf{p}_{\mathrm{n_s}})\right]\in\mathbb{R}^{\mathrm{n_{qp}d^2}\times\mathrm{n_s}}, \\
&\mathrm{POD}\left(\boldsymbol{S}_{\mathbb{B}}\right)\rightarrow\boldsymbol{H}_{\mathbb{B}}=\left[\hat{\boldsymbol{\phi}}_1^{\mathbb{B}},\ldots,\hat{\boldsymbol{\phi}}_{\mathrm{n}_{\mathbb{B}}}^{\mathbb{B}}\right]\in\mathbb{R}^{\mathrm{n_{qp}d}\times\mathrm{n}_{\mathbb{B}}} \quad \text{with } \boldsymbol{S}_{\mathbb{B}}=\left[\hat{\mathbb{B}}(\mathbf{p}_1),\ldots,\hat{\mathbb{B}}(\mathbf{p}_{\mathrm{n_s}})\right]\in\mathbb{R}^{\mathrm{n_{qp}d}\times\mathrm{n_s}}.
\end{aligned} \tag{38}$$

The column vectors $\left\{\hat{\boldsymbol{P}}_i\right\}_{i=1}^{\mathrm{n_s}}$ and $\left\{\hat{\mathbb{B}}_i\right\}_{i=1}^{\mathrm{n_s}}$ contain the components of $\boldsymbol{P}$ and $\mathbb{B}$ at the $\mathrm{n_{qp}}$ quadrature points of the FE model.

By introducing (37) to (36) and recalling that the coefficients in (37) do not depend on the position of the elements in the reduced meshes, we obtain alternative errors

$$\begin{aligned}
{}^{\boldsymbol{u}}\check{e}_{ij} &= \sum_{e\in\mathcal{E}_{\boldsymbol{u}}}\omega_e^{\boldsymbol{u}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_i^{\boldsymbol{u}}:\boldsymbol{\phi}_j^{\boldsymbol{P}}\,\mathrm{d}V-\sum_{e=1}^{\mathrm{M}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_i^{\boldsymbol{u}}:\boldsymbol{\phi}_j^{\boldsymbol{P}}\,\mathrm{d}V \\
\text{and} \quad {}^{\mathbb{y}}\check{e}_{ij} &= \sum_{e\in\mathcal{E}_{\mathbb{y}}}\omega_e^{\mathbb{y}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\varphi_i^{\mathbb{y}}\cdot\boldsymbol{\phi}_j^{\mathbb{B}}\,\mathrm{d}V-\sum_{e=1}^{\mathrm{M}}\int_{\Omega_e}\nabla_{\boldsymbol{X}}\varphi_i^{\mathbb{y}}\cdot\boldsymbol{\phi}_j^{\mathbb{B}}\,\mathrm{d}V.
\end{aligned} \tag{39}$$

For the details on minimisation of (39) in order to obtain the reduced meshes and the weights, the interested reader is referred to Appendix C or [29]. In contrast to the method put forward in [29], the

EC introduced here uses elements instead of single Gauss points, resembling the ECSW method [28]. By doing so, the effective number of quadrature points employed in the reduced model increases, but the implementation is less code invasive.

The linearisation of the weak form of the EC hyper-reduced model (35) becomes

$$\begin{bmatrix} \sum\limits_{e\in\mathcal{E}^{\boldsymbol{u}}} \omega_e^{\boldsymbol{u}}\, {}^e\boldsymbol{B}_{\boldsymbol{u}}^\top\, {}^e\boldsymbol{K}_k^{\boldsymbol{u}\boldsymbol{u}}\, {}^e\boldsymbol{B}_{\boldsymbol{u}} & \sum\limits_{e\in\mathcal{E}^{\boldsymbol{u}}} \omega_e^{\boldsymbol{u}}\, {}^e\boldsymbol{B}_{\boldsymbol{u}}^\top\, {}^e\boldsymbol{K}_k^{\boldsymbol{u}\mathrm{y}}\, {}^e\boldsymbol{B}_{\mathrm{y}} \\ \sum\limits_{e\in\mathcal{E}^{\mathrm{y}}} \omega_e^{\mathrm{y}}\, {}^e\boldsymbol{B}_{\mathrm{y}}^\top\, {}^e\boldsymbol{K}_k^{\mathrm{y}\boldsymbol{u}}\, {}^e\boldsymbol{B}_{\boldsymbol{u}} & \sum\limits_{e\in\mathcal{E}^{\mathrm{y}}} \omega_e^{\mathrm{y}}\, {}^e\boldsymbol{B}_{\mathrm{y}}^\top\, {}^e\boldsymbol{K}_k^{\mathrm{y}\mathrm{y}}\, {}^e\boldsymbol{B}_{\mathrm{y}} \end{bmatrix} \begin{bmatrix} \Delta\tilde{\boldsymbol{u}}_k^r \\ \Delta\tilde{\mathrm{y}}_k^r \end{bmatrix} = - \begin{bmatrix} \sum\limits_{e\in\mathcal{E}^{\boldsymbol{u}}} \omega_e^{\boldsymbol{u}}\, {}^e\boldsymbol{B}_{\boldsymbol{u}}^\top\, {}^e\hat{\boldsymbol{R}}_k^{\boldsymbol{u}} \\ \sum\limits_{e\in\mathcal{E}^{\mathrm{y}}} \omega_e^{\mathrm{y}}\, {}^e\boldsymbol{B}_{\mathrm{y}}^\top\, {}^e\hat{\boldsymbol{R}}_k^{\mathrm{y}} \end{bmatrix}, \tag{40}$$

where ${}^e\boldsymbol{B}_{\boldsymbol{u}}$ and ${}^e\boldsymbol{B}_{\mathrm{y}}$ are the restrictions of $\boldsymbol{B}_{\boldsymbol{u}}$ and $\boldsymbol{B}_{\boldsymbol{u}}$ to the finite element Ω_e. For EC, the reduced system matrix has the same properties, e.g. symmetry and positive definiteness, as the system matrix of the FE model, as the weights are strictly positive. This property is not shared by the collateral basis methods.

The cost for evaluating the constitutive law in the elements of $\mathcal{E}_{\boldsymbol{u}} \cup \mathcal{E}_{\mathrm{y}}$ is roughly $\mathcal{O}(\mathrm{N_{eval}n} + \mathrm{n_{qp}^{el}m})$, where m is the number of elements in the union of the reduced meshes and $\mathrm{N_{eval}}$ the number of associated DoFs. The assembly of the residuum and the tangent matrix is proportional to $\mathcal{O}(\mathrm{nN_{eval}})$ and $\mathcal{O}(\mathrm{n^2N_{eval}} + \mathrm{nN_{eval}})$, respectively.

4.5. Reduced Integration Domain

For RID, two reduced integration domains $\Omega_{\mathrm{RID}}^{\boldsymbol{u}} \subset \mathcal{T}$ and $\Omega_{\mathrm{RID}}^{\mathrm{y}} \subset \mathcal{T}$ are introduced, which are used to define test functions $\delta\tilde{\boldsymbol{u}}$ and $\delta\tilde{\mathrm{y}}$ with support only in $\Omega_{\mathrm{RID}}^{\boldsymbol{u}}$ and $\Omega_{\mathrm{RID}}^{\mathrm{y}}$. Hence, the nonlinearities will be computed solely in $\Omega_{\mathrm{RID}}^{\boldsymbol{u}}$ and $\Omega_{\mathrm{RID}}^{\mathrm{y}}$, which provides the desired reduction of computational cost.

In the discrete setting, the test functions in $\boldsymbol{B}_{\mathrm{RID}}$ with confined support are expressed in terms of the reduced bases $\boldsymbol{B}$ as

$$\begin{aligned} \boldsymbol{B}_{\mathrm{RID}} &:= \begin{bmatrix} \mathbf{P}_{\mathrm{RID}}^{\boldsymbol{u}} \left(\mathbf{P}_{\mathrm{RID}}^{\boldsymbol{u}}\right)^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\mathrm{RID}}^{\mathrm{y}} \left(\mathbf{P}_{\mathrm{RID}}^{\mathrm{y}}\right)^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_{\boldsymbol{u}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{B}_{\mathrm{y}} \end{bmatrix} = \mathbf{P}_{\mathrm{RID}} \mathbf{P}_{\mathrm{RID}}^\top \boldsymbol{B} \\ &\text{with } \mathbf{P}_{\mathrm{RID}}^{\boldsymbol{u}} = \left[\boldsymbol{e}_{\rho_1^{\boldsymbol{u}}}, \ldots, \boldsymbol{e}_{\rho_{\mathrm{l}_{\boldsymbol{u}}}^{\boldsymbol{u}}} \right] \in \mathbb{N}^{\mathrm{N}_{\boldsymbol{u}} \times \mathrm{l}_{\boldsymbol{u}}},\ \mathbf{P}_{\mathrm{RID}}^{\mathrm{y}} = \left[\boldsymbol{e}_{\rho_1^{\mathrm{y}}}, \ldots, \boldsymbol{e}_{\rho_{\mathrm{l}_{\mathrm{y}}}^{\mathrm{y}}} \right] \in \mathbb{N}^{\mathrm{N}_{\mathrm{y}} \times \mathrm{l}_{\mathrm{y}}} \\ &\text{and } \mathbf{P}_{\mathrm{RID}} := \begin{bmatrix} \mathbf{P}_{\mathrm{RID}}^{\boldsymbol{u}} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\mathrm{RID}}^{\mathrm{y}} \end{bmatrix} \in \mathbb{N}^{\mathrm{N} \times \mathrm{l}}, \end{aligned} \tag{41}$$

where $\boldsymbol{e}_{\rho_i^{\boldsymbol{u}}} \in \mathbb{R}^{\mathrm{N}_{\boldsymbol{u}}}$ for $i = 1, \ldots, \mathrm{l}_{\boldsymbol{u}}$ and $\boldsymbol{e}_{\rho_j^{\mathrm{y}}} \in \mathbb{R}^{\mathrm{N}_{\boldsymbol{u}}}$ for $j = 1, \ldots, \mathrm{l}_{\mathrm{y}}$ are unit vectors with only one non-zero component in the $\rho_i^{\boldsymbol{u}}$-th and ρ_j^{y}-th entry. The indices $\left\{\rho_i^{\boldsymbol{u}}\right\}_{i=1}^{\mathrm{l}_{\boldsymbol{u}}}$ and $\left\{\rho_i^{\mathrm{y}}\right\}_{i=1}^{\mathrm{l}_{\mathrm{y}}}$ are the interior DoFs of $\Omega_{\mathrm{RID}}^{\boldsymbol{u}}$ and $\Omega_{\mathrm{RID}}^{\mathrm{y}}$.

The reduced domains are generated from the gradients of the reduced bases

$$\boldsymbol{H}_{\nabla_{\boldsymbol{X}}\boldsymbol{u}} := \left[\nabla_{\boldsymbol{X}} \hat{\boldsymbol{\varphi}}_1^{\boldsymbol{u}}, \ldots, \nabla_{\boldsymbol{X}} \hat{\boldsymbol{\varphi}}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} \right] \in \mathbb{R}^{\mathrm{n_{qp}d^2} \times \mathrm{n}_{\boldsymbol{u}}} \quad \text{and} \quad \boldsymbol{H}_{\nabla_{\boldsymbol{X}}\mathrm{y}} := \left[\nabla_{\boldsymbol{X}} \hat{\boldsymbol{\varphi}}_1^{\mathrm{y}}, \ldots, \nabla_{\boldsymbol{X}} \hat{\boldsymbol{\varphi}}_{\mathrm{n}_{\mathrm{y}}}^{\mathrm{y}} \right] \in \mathbb{R}^{\mathrm{n_{qp}d} \times \mathrm{n}_{\mathrm{y}}}. \tag{42}$$

The columns $\left\{\nabla_{\boldsymbol{X}} \hat{\boldsymbol{\varphi}}_i^{\boldsymbol{u}}\right\}_{i=1}^{\mathrm{n}_{\boldsymbol{u}}}$ and $\left\{\nabla_{\boldsymbol{X}} \hat{\boldsymbol{\varphi}}_i^{\mathrm{y}}\right\}_{i=1}^{\mathrm{n}_{\mathrm{y}}}$ contain the components of the gradients of $\left\{\boldsymbol{\varphi}_i^{\boldsymbol{u}}\right\}_{i=1}^{\mathrm{n}_{\boldsymbol{u}}}$ and $\left\{\varphi_i^{\mathrm{y}}\right\}_{i=1}^{\mathrm{n}_{\mathrm{y}}}$ in the $\mathrm{n_{qp}}$ quadrature points of the FE model. Applying the DEIM Algorithm 1 to $\boldsymbol{H}_{\nabla_{\boldsymbol{X}}\boldsymbol{u}}$ and $\boldsymbol{H}_{\nabla_{\boldsymbol{X}}\mathrm{y}}$ returns the indices used to construct the reduced domains $\Omega_{\mathrm{RID}}^{\boldsymbol{u}}$ and $\Omega_{\mathrm{RID}}^{\mathrm{y}}$. Strictly speaking, the reduced domains are the unions of elements containing these indices.

Using (41) in (16) renders the discrete weak from of the RID hyper-reduced model

$$\boldsymbol{R}_k^r := \boldsymbol{B}^\top \mathbf{P}_{\mathrm{RID}} \mathbf{P}_{\mathrm{RID}}^\top \hat{\boldsymbol{R}}_k = \mathbf{0} \tag{43}$$

and its linearisation becomes

$$\underbrace{\boldsymbol{B}^{\top}\mathbf{P}_{\mathrm{RID}}}_{\text{precomputed: } \mathbb{R}^{\mathrm{n}\times\mathrm{l}}}\mathbf{P}_{\mathrm{RID}}^{\top}\boldsymbol{K}_k\boldsymbol{B}\Delta\tilde{\boldsymbol{y}}_k^r = -\underbrace{\boldsymbol{B}^{\top}\mathbf{P}_{\mathrm{RID}}}_{\text{precomputed: } \mathbb{R}^{\mathrm{n}\times\mathrm{l}}}\mathbf{P}_{\mathrm{RID}}^{\top}\hat{\boldsymbol{R}}_k. \tag{44}$$

The cost for evaluating the constitutive law is roughly $\mathcal{O}(\mathrm{N_{eval}n} + \mathrm{n_{qp}^{el}m})$, where m is the number of elements of the union $\Omega_{\mathrm{RID}}^{\boldsymbol{u}} \cup \Omega_{\mathrm{RID}}^{\mathrm{y}}$ and $\mathrm{N_{eval}}$ the number of DoFs associated with those elements. The approximate costs of assembling the residuum and tangent stiffness matrix are $\mathcal{O}(\mathrm{lN_{eval}})$ and $\mathcal{O}(\mathrm{nlN_{eval}} + \mathrm{lN_{eval}})$, respectively.

For (43) to be a well-posed problem, $\boldsymbol{B}_{\mathrm{RID}}$ is required to have full column rank. If that is not fulfilled or the accuracy of the model is poor, the l element layers surrounding $\Omega_{\mathrm{RID}}^{\boldsymbol{u}}$ and $\Omega_{\mathrm{RID}}^{\mathrm{y}}$ are included (cf. Algorithm 4 in Appendix D).

5. Numerical Results

5.1. Test Problem

The magneto-mechanical material model chosen for the numerical studies is of Neo-Hookean type

$$\begin{aligned}\Psi(\boldsymbol{F},\mathbb{H}) =& \frac{1}{2}\lambda_2\left[\boldsymbol{F}:\boldsymbol{F} - \mathrm{d} - 2\ln J\right] + \frac{1}{2}\lambda_1\ln^2 J - \frac{1}{2}\mu J\mathbb{H}\cdot\boldsymbol{C}^{-1}\cdot\mathbb{H}\\ &\text{using } J = \det\boldsymbol{F} \text{ and } \boldsymbol{C} = \boldsymbol{F}^{\top}\cdot\boldsymbol{F},\end{aligned} \tag{45}$$

combining isotropic elastic with linear isotropic magnetic material properties. For further details including expressions of the Piola stress $\boldsymbol{P}$ and the magnetic induction $\mathbb{B}$, see [41].

The mesh used for all numerical tests is displayed in Figure 1, where 1840 quadratic finite elements are used to discretise the continuum body $\mathcal{B}_0$, resulting in $\mathrm{N} = \mathrm{N}_{\boldsymbol{u}} + \mathrm{N}_{\mathrm{y}} = 14{,}882 + 7441 = 22{,}323$ DoFs. For numerical integration, a 4×4 Gaussian Quadrature is employed. The implementation of the tests is based on the open–source FE library deal.II [47].

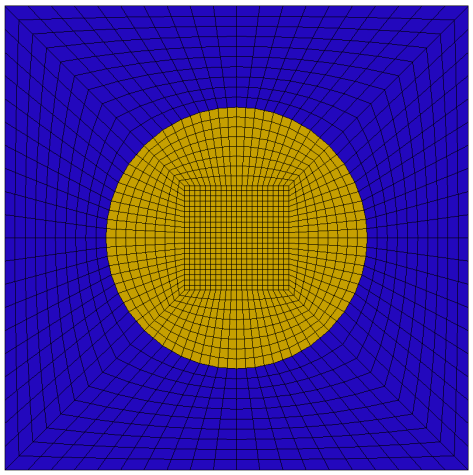


Figure 1. FE mesh of a Unit Cell discretised using $\mathrm{M} = 1840$ elements with quadratic Ansatz functions resulting in $\mathrm{N}_{\boldsymbol{u}} = 14{,}882$ and $\mathrm{N}_{\mathrm{y}} = 7441$ DoFs.

In Table 1, the dimensionless Lamé parameters λ_1 and λ_2 and magnetic permeability μ are given. The inclusion/particle has ten times stronger material parameters than the matrix.

Table 1. Material parameters.

	Matrix	**Inclusion**
λ_1	12	120
λ_2	8	80
μ	0.001	0.01

Additionally, the parameter domain for the two-dimensional problem to be investigated is prescribed by

$$\mathcal{P} = \underbrace{(0.9, 1.2)}_{\overline{\boldsymbol{F}}[1,1]} \times \underbrace{(-0.2, 0.2)}_{\overline{\boldsymbol{F}}[1,2]} \times \underbrace{(-0.2, 0.2)}_{\overline{\boldsymbol{F}}[2,1]} \times \underbrace{(0.9, 1.2)}_{\overline{\boldsymbol{F}}[2,2]} \times \underbrace{(-10, 10)}_{\overline{\mathbb{H}}[1]} \times \underbrace{(-10, 10)}_{\overline{\mathbb{H}}[2]} \subset \mathbb{R}^6. \tag{46}$$

The output of interest from the reduced-models are the homogenised Piola stress $\overline{\boldsymbol{P}}$ and magnetic induction $\overline{\mathbb{B}}$, for which the relative error measures

$$\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}) = \mathrm{median}\left\{ \frac{\left\| \overline{\boldsymbol{P}}_i^{\mathrm{FOM}} - \overline{\boldsymbol{P}}_i^{\mathrm{ROM}} \right\|_F}{\left\| \overline{\boldsymbol{P}}_i^{\mathrm{FOM}} \right\|_F} \right\}_{i=1}^{|\mathcal{V}|} \quad \text{and} \quad \mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}) = \mathrm{median}\left\{ \frac{\left\| \overline{\mathbb{B}}_i^{\mathrm{FOM}} - \overline{\mathbb{B}}_i^{\mathrm{ROM}} \right\|_2}{\left\| \overline{\mathbb{B}}_i^{\mathrm{FOM}} \right\|_2} \right\}_{i=1}^{|\mathcal{V}|} \tag{47}$$

are defined for any set of validation parameters $\mathcal{V}$.

To validate the accuracy and robustness of the reduced models, two sets of randomly chosen parameters

$$\mathcal{V}_{\mathrm{I}} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{200}\} \subset \mathcal{P} \quad \text{and} \quad \mathcal{V}_{\mathrm{II}} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{2000}\} \subset \mathcal{P} \tag{48}$$

are defined and used in combination with (47).

The computation of $\overline{\boldsymbol{P}}^{\mathrm{ROM}}$ and $\overline{\mathbb{B}}^{\mathrm{ROM}}$ (8) requires $\tilde{\boldsymbol{u}}$ and $\tilde{\mathbb{y}}$ to be computed in all cells of the FE mesh using (9). This can be done more efficiently using an auxiliary basis for $\boldsymbol{P}$ and $\mathbb{B}$ together with gappy reconstruction, but this renders an additional error. As our focus is on the numerical study of the performance of the hyper-reduction methods, $\tilde{\boldsymbol{u}}$ and $\tilde{\mathbb{y}}$ are computed for the whole mesh and the constitutive law is subsequently employed to obtain $\overline{\boldsymbol{P}}^{\mathrm{ROM}}$ and $\overline{\mathbb{B}}^{\mathrm{ROM}}$.

It is established in the field of computational homogenisation that the application of linear boundary conditions overestimates the energy compared to e.g. periodic boundary conditions, in particular for small RVEs like the Unit Cell. This has been investigated extensively in [40] for magneto-mechanics. As the choice of boundary conditions does not affect the hyper-reduction methods, the findings from the numerical studies are expected to be valid for different types of boundary conditions. Therefore, due to their simplicity, linear boundary conditions have been chosen to carry out the numerical studies.

5.2. Validation of Galerkin ROM

In order to construct the reduced basis, the parameter space $\mathcal{P}$ has to be sampled. As the number of sampling points increases exponentially with the dimension of the parameter domain for full tensor grids, sparse grids [48,49] are employed. Sparse grids are based on one-dimensional quadrature rules and a sparse tensor product, which alleviates the curse of dimensionality. For that reason, sparse grids are frequently used in sampling, interpolation and integration of high dimensional functions.

In Figure 2, the sampling of the unit square using a full tensor grid and sparse grids is displayed. Sparse grids built from the one-dimensional Gauss–Legendre quadrature are used to sample the six-dimensional parameter domain $\mathcal{P}$ (46).

As the hyper-reduced models are built on top of an existing reduced basis, the accuracy of the Galerkin reduced model (16) for varying cardinalities $\mathrm{n}_{\boldsymbol{u}}$ and $\mathrm{n}_{\mathbb{y}}$ of the reduced bases for the fluctuation

fields is displayed in Figure 3. For increasing $n_{\boldsymbol{u}}$ and n_y, the errors $\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_I)$ and $\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_I)$ decrease monotonously, though the impact of n_y on $\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_I)$ becomes negligible for $n_y \geq 10$. The reduced basis was constructed from $n_s = 4541$ training simulations. As a compromise between accuracy and efficiency, a reduced basis with $n_{\boldsymbol{u}} = 20$ and $n_y = 10$ is chosen for all following numerical studies.

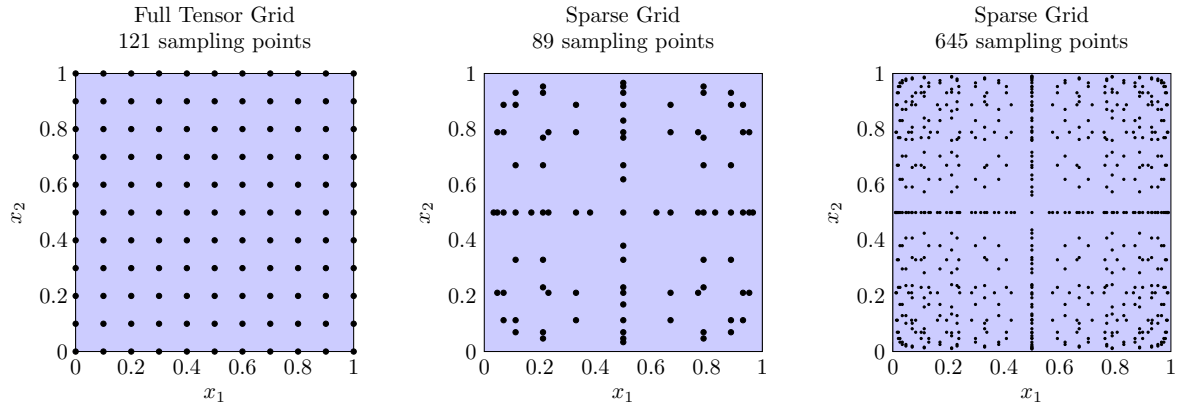


Figure 2. Sampling of the two-dimensional parameter domain $[0,1] \times [0,1]$ using a full tensor grid and two sparse grids based on a one-dimensional Gauss–Legendre quadrature with different sampling densities.

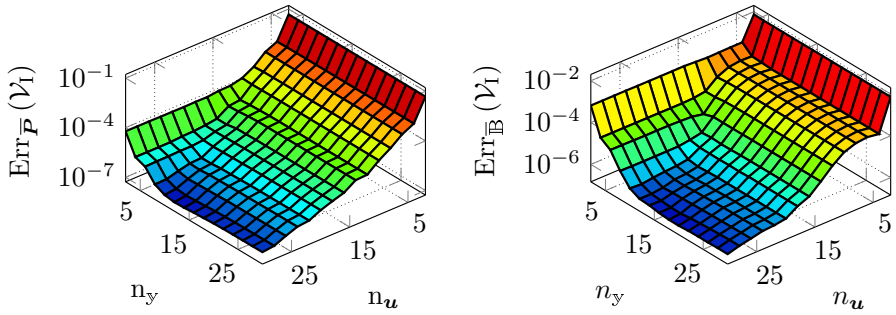


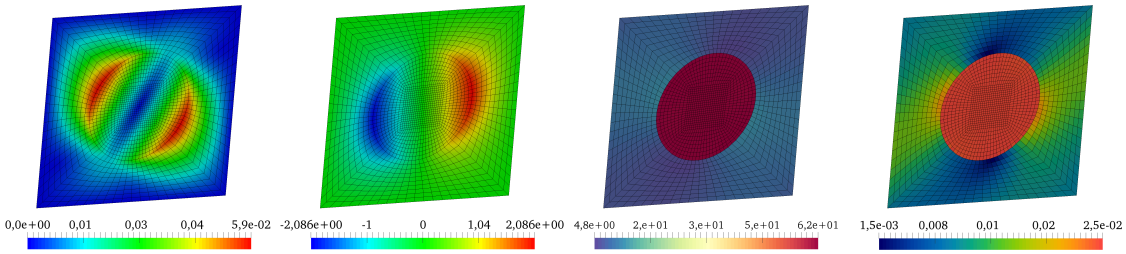
Figure 3. Errors in homogenised quantities of interest for varying sizes of reduced bases $n_{\boldsymbol{u}}$ and n_y computed from $n_s = 4541$ training simulations.

The error of the Galerkin ROM is not only caused by the truncation of the POD basis but also by an insufficient selection of training parameters $\mathcal{S}$. In this case, insufficient refers to a too sparse sampling of the parameter domain $\mathcal{P}$. In Table 2, the errors $\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_I)$ and $\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_I)$ for $n_{\boldsymbol{u}} = 20$ and $n_y = 10$ for three different training sets are given. The training set $\mathcal{S}$ with $n_s = 4541$ is considered sufficiently large as the errors for the two sets with greater cardinality are not significantly smaller.

Table 2. Output error for different numbers of training parameters n_s for a reduced basis of fixed size $n_{\boldsymbol{u}} = 20$ and $n_y = 10$.

$\lvert\mathcal{S}\rvert$	**4541**	**12,841**	**33,193**
$\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_I)$	2.75×10^{-6}	2.59×10^{-6}	2.53×10^{-6}
$\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_I)$	1.55×10^{-6}	1.51×10^{-6}	1.62×10^{-6}

The results of a ROM for one element in $\mathcal{V}_I$ are displayed in Figure 4. The errors in the homogenised quantities are small $\mathcal{O}(10^{-6})$ and differences between the ROM and FOM in the primary fields $\tilde{\boldsymbol{u}}$ and $\tilde{y}$ can not be seen with the unaided eye. Due to the inclusions ten times larger mechanical and magnetic material parameters, the Piola stress and magnetic induction inside the inclusion are significantly larger.



a) $\|\tilde{\boldsymbol{u}}\|_2$ b) $\tilde{\mathrm{y}}$ c) $\|\boldsymbol{P}\|_F$ d) $\|\mathbb{B}\|_2$

Figure 4. Results of a reduced model for $\overline{\boldsymbol{F}} = \begin{bmatrix} 1.132 & 0.182 \\ 0.144 & 1.145 \end{bmatrix}$ and $\overline{\mathbb{H}} = \begin{bmatrix} -8.231 \\ 0.129 \end{bmatrix}$ rendering $\overline{\boldsymbol{P}}^{\mathrm{ROM}} = \begin{bmatrix} 5.875 & 2.943 \\ 2.756 & 6.026 \end{bmatrix}$ and $\overline{\mathbb{H}}^{\mathrm{ROM}} = \begin{bmatrix} -1.123 \times 10^{-2} \\ 3.520 \times 10^{-3} \end{bmatrix}$ using a reduced basis with $\mathrm{n}_{\boldsymbol{u}} = 20$ and $\mathrm{n}_{\mathrm{y}} = 10$. The norm of the displacement fluctuations (**a**) the potential fluctuations; (**b**) the Frobenius norm of the element averaged Piola stress; (**c**) and the norm of the element averaged magnetic induction; (**d**) are depicted in the deformed Unit Cell. The errors for the homogenised quantities $\frac{\|\overline{\boldsymbol{P}}^{\mathrm{FOM}} - \overline{\boldsymbol{P}}^{\mathrm{ROM}}\|_F}{\|\overline{\boldsymbol{P}}^{\mathrm{FOM}}\|_F} = 3.4 \times 10^{-6}$ and $\frac{\|\overline{\mathbb{B}}^{\mathrm{FOM}} - \overline{\mathbb{B}}^{\mathrm{ROM}}\|_2}{\|\overline{\mathbb{B}}^{\mathrm{FOM}}\|_2} = 3.1 \times 10^{-6}$ are small. For better quality, we point to the online version of the article.

5.3. DEIM

Based on a reduced basis with $\mathrm{n}_{\boldsymbol{u}} = 20$ and $\mathrm{n}_{\mathrm{y}} = 10$, the results of the DEIM hyper-reduced model (23) are summarised in Figure 5, where the collateral basis $\boldsymbol{H}_{\boldsymbol{R}}$ is computed based on residua collected during the solution of (16) for $\mathrm{n}_{\mathrm{s}} = 4541$ training parameters. For the POD computations, only residua fulfilling $\left\|\hat{\boldsymbol{R}}^{\boldsymbol{u}}_j(\mathrm{p}_i)\right\| / \left\|\hat{\boldsymbol{R}}^{\boldsymbol{u}}_1(\mathrm{p}_i)\right\|, \left\|\hat{\boldsymbol{R}}^{\mathrm{y}}_j(\mathrm{p}_i)\right\| / \left\|\hat{\boldsymbol{R}}^{\mathrm{y}}_1(\mathrm{p}_i)\right\| > 10^{-4}$ with $i \in [1, 4541]$ and the iteration index j were taken into account.

It is well-established that DEIM models lack robustness for nonlinear pPDEs [26], which is exposed in Figure 5a. There is just a small region of combinations of numbers of collateral reduced basis functions $(\mathrm{r}_{\boldsymbol{u}}, \mathrm{r}_{\mathrm{y}})$, for which the reduced model converged for all parameters in the validation set $\mathcal{V}_{\mathrm{I}}$. There are two possible causes that prevent the convergence of a model. The first is the occurrence of unphysical deformations expressed by $\det \boldsymbol{F} \leq 0$ during the solution process and the second is an insufficient reduction of the residuum $\boldsymbol{R}^r_{\max} / \boldsymbol{R}^r_0 > 10^{-6}$, where max = 10 is the highest admissible number of nonlinear solver iterations. From the 135,200 solutions of (23) for 676 different combinations of $(\mathrm{r}_{\boldsymbol{u}}, \mathrm{r}_{\mathrm{y}})$ computed in this study, 11,992 failed to converge. It is remarkable that larger $\mathrm{r}_{\boldsymbol{u}}$ and r_{y} do not result in more robust models. Figures 5b,c show the output errors, where we have to note that, for the calculation of $\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_{\mathrm{I}})$ and $\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_{\mathrm{II}})$, only the converged runs are taken into account. The errors decrease with increasing $\mathrm{r}_{\boldsymbol{u}}$ and r_{y} but certainly not monotonously. For a ROM with $\mathrm{r}_{\boldsymbol{u}} = 37$ and $\mathrm{r}_{\mathrm{y}} = 25$, small errors $\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_{\mathrm{I}}) = 1.2 \times 10^{-3}$ and $\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_{\mathrm{I}}) = 5.5 \times 10^{-4}$ are obtained with one simulation failing to converge. It is not reasonable to use greater $\mathrm{r}_{\boldsymbol{u}}$ and r_{y} as the achievable reduction of the errors is disproportionate to the increasing numerical cost of the ROM.

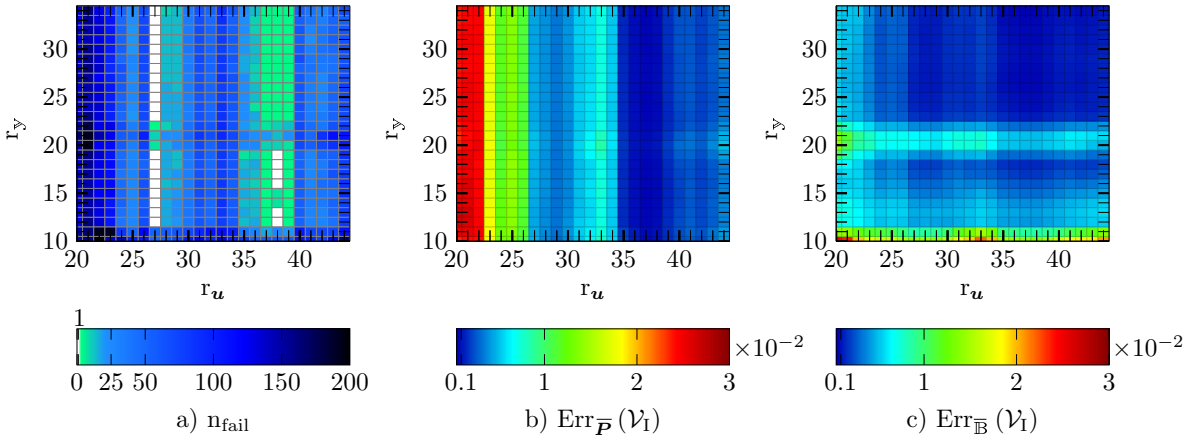


Figure 5. (**a**) robustness and (**b**,**c**) output error analysis for varying numbers of DEIM indices r_u and r_y for a reduced basis of size $n_u = 20$ and $n_y = 10$.

5.4. Gappy POD

For the Gappy POD study, the same collateral basis $\boldsymbol{H_R}$ as in the DEIM study is used. In order to have a fair comparison with DEIM, two sets of gappy points, one for the approximation of $\hat{\boldsymbol{R}}^u$ and the other for $\hat{\boldsymbol{R}}^y$ in (28), are determined by applying Algorithm 2 separately to the collateral bases.

In Figure 6, the robustness and accuracy of Gappy POD is studied for different combinations of (r_u, r_y). To facilitate an adequately accurate approximation of the nonlinearities, large enough numbers of gappy points (p_u, p_y) are employed. Figure 6a shows that linear regression improves the robustness as more combinations of (r_u, r_y) exhibit no convergence failures compared to DEIM (c.f. Figure 5). Nonetheless, 11,339 out of 135,200 simulation runs failed, mostly for smaller values of r_u and r_y. For the failed runs, either the condition $\boldsymbol{R}^r_{10}/\boldsymbol{R}^r_0 < 10^{-6}$ is not fulfilled after ten iterations or unphysical deformations $\det \boldsymbol{F} \leq 0$ are predicted.

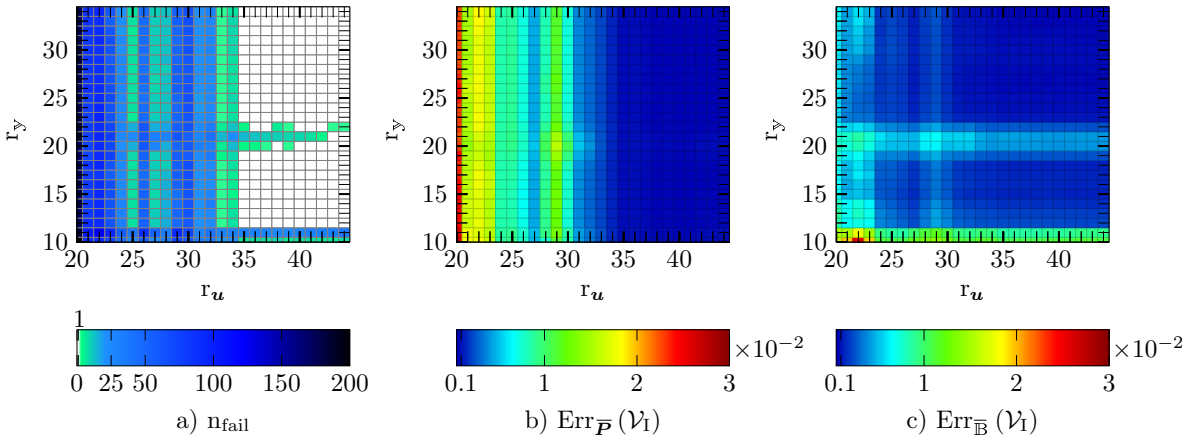


Figure 6. (**a**) robustness and (**b**,**c**) output error analysis for varying numbers of residuum modes r_u and r_y for sufficiently large numbers of gappy points $p_u = \lceil r_u/2 + 10 \rceil$ and $p_y = r_y + 20$ for a reduced basis of size $n_u = 20$ and $n_y = 10$.

For $r_u = 36$ and $r_y = 24$, a combination yielding supposedly robust and accurate models is chosen to investigate the errors for varying numbers of gappy points (p_u, p_y) (see Figure 7). Except for $p_u \leq 20$, no robustness deficiencies occur. The error $Err_{\overline{P}}(\mathcal{V}_I)$ decreases with increasing p_u and is hardly affected by changes of p_y. Similarly, $Err_{\overline{\mathbb{B}}}(\mathcal{V}_I)$ reduces for larger p_y and is only minorly affected by p_u. For a reduced model using $p_u = 50$ and $p_y = 58$ gappy points, the errors $Err_{\overline{P}}(\mathcal{V}_I) = 3.6 \times 10^{-4}$ and $Err_{\overline{\mathbb{B}}}(\mathcal{V}_I) = 3.0 \times 10^{-4}$ were achieved.

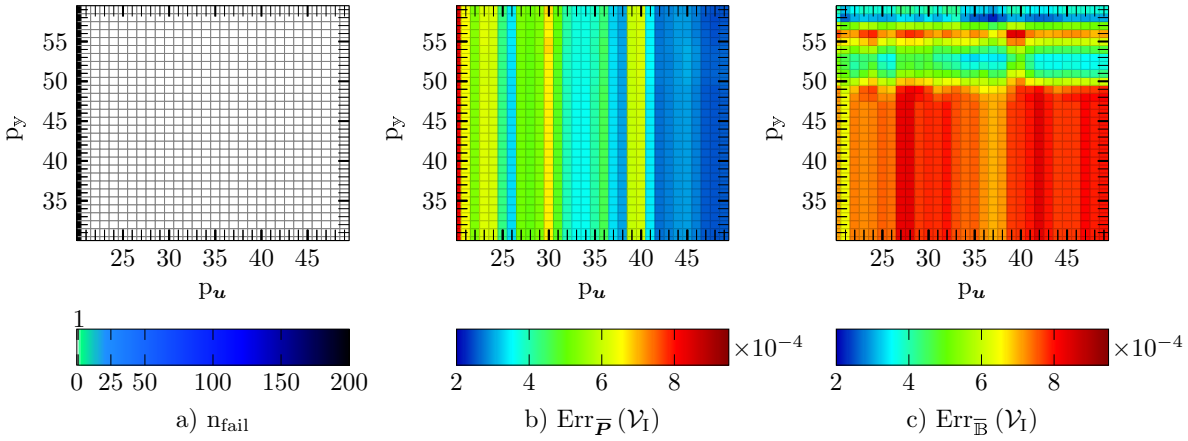


Figure 7. Influence of gappy point numbers p_u and p_y on (**a**) robustness and (**b**,**c**) accuracy for a reduced basis of size $n_u = 20$ and $n_y = 10$ using $r_u = 36$ and $r_y = 24$ residuum modes.

5.5. GNAT

For the study of GNAT, the collateral bases contained in $\boldsymbol{H}_{\boldsymbol{R}^u}$ and $\boldsymbol{H}_{\boldsymbol{R}^y}$ are constructed by gathering the residua from the solution of (32) and a subsequent application of POD. Only residua fulfilling $\left\|\hat{R}^u_j(\mathbf{p}_i)\right\| / \left\|\hat{R}^u_1(\mathbf{p}_i)\right\|, \left\|\hat{R}^y_j(\mathbf{p}_i)\right\| / \left\|\hat{R}^y_1(\mathbf{p}_i)\right\| > 5 \times 10^{-3}$ with $i \in [1, 4541]$ and the iteration index j were taken into account. The gappy points are obtained by applying Algorithm 2 separately to the collateral mechanical and magnetic basis.

Similarly to the previous studies, the robustness and accuracy of GNAT is tested for different combinations of (r_u, r_y) using a large enough number of gappy points. The results are depicted in Figure 8. We never observed unphysical deformations $\det \boldsymbol{F} \leq 0$ for GNAT hyper-reduced models, the unsuccessful runs are due to reduced models failing to sufficiently reduce the residuum $\boldsymbol{R}^r_{15} / \boldsymbol{R}^r_0 < 10^{-3}$. As the Gauss–Newton scheme does not exhibit quadratic convergence, we allow for up to 15 iterations to minimise the residuum. Furthermore, the solution of the nonlinear least squares problem (32) does not render $\boldsymbol{R}^r \equiv \boldsymbol{0}$ in general and consequently the convergence criterion is set to 10^{-3}. While the error $\mathrm{Err}_{\overline{P}}(\mathcal{V}_I)$ in Figure 8b reduces with increasing r_u and values in the order of 10^{-3} can be achieved, the error $\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_I)$ in Figure 8c increases for greater r_u up to a certain point and $\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_I) \approx 5 \times 10^{-2}$ seems feasible.

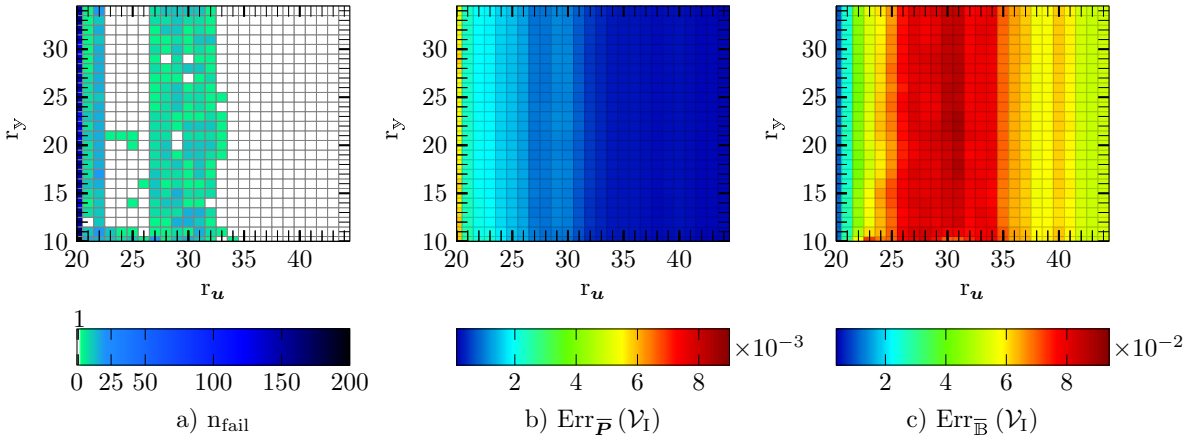


Figure 8. (**a**) robustness and (**b**,**c**) output error analysis for varying numbers of residuum modes r_u and r_y for sufficiently large numbers of gappy points $p_u = \lceil r_u/2 + 10 \rceil$ and $p_y = r_y + 20$ for a reduced basis of size $n_u = 20$ and $n_y = 10$.

5.6. Empirical Cubature

In Figure 9, the accuracy of EC hyper-reduced models for different numbers of elements in the reduced mesh is depicted. To construct the EC model 4541 snapshots of $\boldsymbol{P}$ and $\mathbb{B}$ are taken from the solution of (15) and processed into $n_{\boldsymbol{P}} = 120$ and $n_{\mathbb{B}} = 100$ POD modes. The singular value distribution of the snapshot matrices is depicted in Figure 10, indicating that $n_{\boldsymbol{P}} = 120$ and $n_{\mathbb{B}} = 100$ POD modes are sufficient to represent the stress and induction state.

Thereafter, the weights and elements of the reduced meshes are determined by approximately solving (A3) using Algorithm 3, given in Appendix C. Figure 9a shows that $\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_\mathrm{I})$ decreases for greater $m_{\boldsymbol{u}}$ and is barely affected by m_y, whereas $\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_\mathrm{I})$ decreases for greater values of both $m_{\boldsymbol{u}}$ and m_y, but the dependence on m_y is more pronounced. As EC does not employ collateral bases combined with linear regression or interpolation, no convergence issues occur for the EC hyper-reduced models.

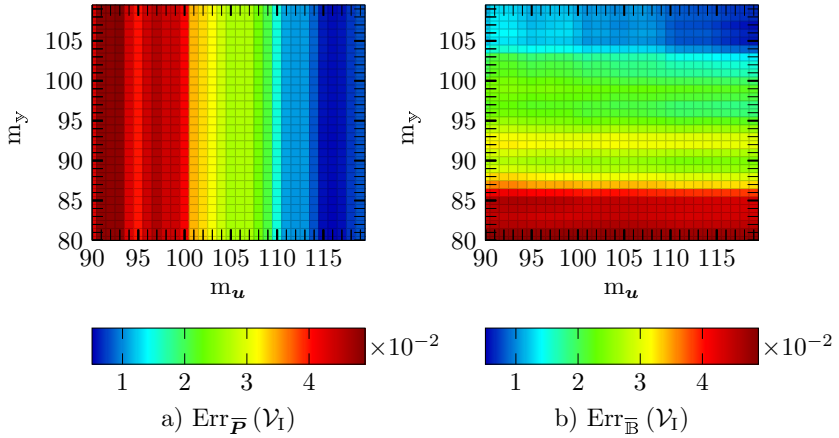


Figure 9. Output error analysis for varying numbers of elements $m_{\boldsymbol{u}}$ and m_y constituting the reduced meshes $\mathcal{E}_{\boldsymbol{u}}$ and $\mathcal{E}_\mathrm{y}$ built using $n_{\boldsymbol{P}} = 120$ stress and $n_{\mathbb{B}} = 100$ induction modes for a reduced basis of size $n_{\boldsymbol{u}} = 20$ and $n_\mathrm{y} = 10$.

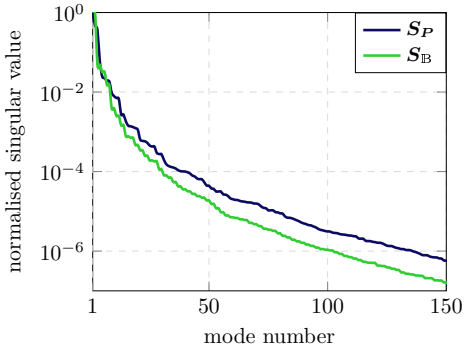
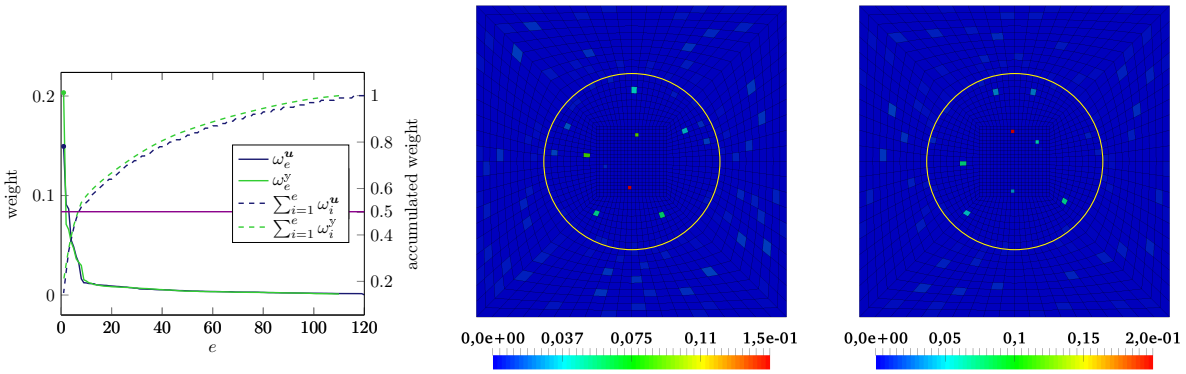


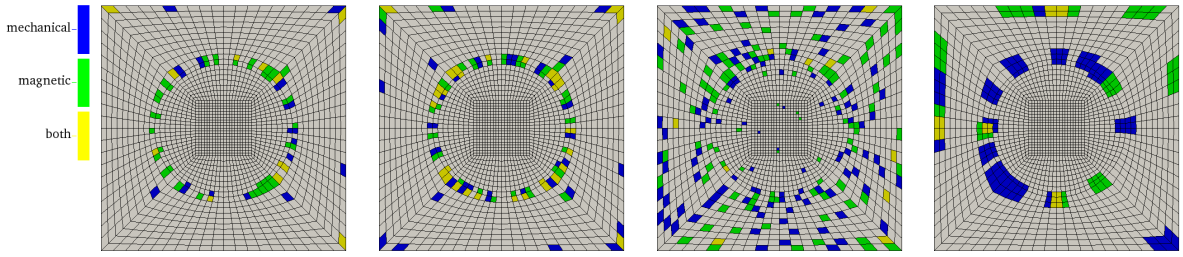
Figure 10. Normalised singular values of the stress $\boldsymbol{S}_{\boldsymbol{P}}$ and induction snapshot matrix $\boldsymbol{S}_{\mathbb{B}}$.

In Figure 11a, the weights used in a reduced mesh are plotted. Only a small number of elements accumulate more than half of the total weight sum and that is the reason why the errors depicted in Figure 9 decrease slowly with increasing $m_{\boldsymbol{u}}$ and m_y. The distribution of elements in the reduced meshes with a focus on elements equipped with relatively large weights is shown in Figures 11b,c. It is remarkable that the elements with large weights are all located inside the inclusion, whereas all the other hyper-reduced models (DEIM, GappyPOD and RID) evaluate the nonlinearities in elements at the boundary or in the vicinity of the interface between matrix and inclusion (c.f. Figure 12).



a) Values of $\boldsymbol{\omega}^{\boldsymbol{u}}$ and $\boldsymbol{\omega}^{\mathrm{y}}$ b) Distribution of $\mathcal{E}_{\boldsymbol{u}}$ in $\mathcal{T}$ coloured by values of $\boldsymbol{\omega}^{\boldsymbol{u}}$ c) Distribution of $\mathcal{E}_{\mathrm{y}}$ in $\mathcal{T}$ coloured by values of $\boldsymbol{\omega}^{\mathrm{y}}$

Figure 11. (**a**) values and distribution of weights (**b**) $\boldsymbol{\omega}^{\boldsymbol{u}}$ and (**c**) $\boldsymbol{\omega}^{\mathrm{y}}$ in $\mathcal{T}$ for reduced meshes $\mathcal{E}_{\boldsymbol{u}}$ and $\mathcal{E}_{\mathrm{y}}$ with $\mathrm{m}_{\boldsymbol{u}} = 120$ and $\mathrm{m}_{\mathrm{y}} = 110$ elements. The yellow circles mark the boundary between matrix and inclusion. For better quality, we point to the online version of the article.



a) DEIM b) Gappy POD c) EC d) RID

Figure 12. Elements in $\mathcal{T}$ relevant for the hyper-reduced models in Table 4.

5.7. Reduced Integration Domain

To construct the reduced integration domain, the gradients of $\{\boldsymbol{\varphi}^{\boldsymbol{u}}\}_{i=1}^{\mathrm{n}_{\boldsymbol{u}}=20}$ and $\{\varphi^{\mathrm{y}}\}_{i=1}^{\mathrm{n}_{\mathrm{y}}=10}$ are computed and the application of Algorithm 4 yields the reduced domains $\Omega^{\boldsymbol{u}}_{\mathrm{RID}}$ and $\Omega^{\mathrm{y}}_{\mathrm{RID}}$. The errors for three different choices of the number of surrounding layers l are listed in Table 3. As expected, the errors decrease with increasing l but solving (43) becomes computationally more expensive. If no surrounding layers are included, the reduced bases $\{\boldsymbol{\varphi}^{\boldsymbol{u}}\}_{i=1}^{\mathrm{n}_{\boldsymbol{u}}=20}$ and $\{\varphi^{\mathrm{y}}\}_{i=1}^{\mathrm{n}_{y}=10}$ in $\Omega^{\boldsymbol{u}}_{\mathrm{RID}}$ and $\Omega^{\mathrm{y}}_{\mathrm{RID}}$ are linearly dependent and consequently (43) is not well-posed.

Table 3. Accuracy of RID depending on the number of neighbouring layers l for $\mathrm{n}_{\boldsymbol{u}} = 20$ and $\mathrm{n}_{\mathrm{y}} = 10$.

l	1	2	3
$\mathrm{Err}_{\overline{P}}(\mathcal{V}_{\mathrm{I}})$	9.5×10^{-4}	3.3×10^{-4}	1.5×10^{-4}
$\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_{\mathrm{I}})$	8.0×10^{-4}	3.2×10^{-4}	1.8×10^{-4}

As for EC, no convergence problems have been observed for RID hyper-reduced models.

5.8. Comparison of the Hyper-Reduction Methods

In Table 4, the performance statistics of reduced models for each hyper-reduction method except GNAT are provided. The parameters for the models, which are listed in Table 5, were chosen based on the results from the previous sections to achieve high accuracy at moderate computational cost. We except GNAT from the comparison as these reduced models are inferior to Gappy POD models with respect to robustness and accuracy. Most importantly, the error $\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V})$ produced by GNAT models is too large.

The number of elements in which either the mechanical or magnetic nonlinearity have to be computed are denoted by $m_{\boldsymbol{u}}$ and m_{y}, respectively. The aforementioned elements are highlighted in Figure 12. It is possible to use the same elements for the evaluation of the nonlinearities for all hyper-reduction methods except DEIM, rendering supposedly slightly less accurate but more efficient reduced models. However, for the sake of comparison of all introduced Hyper-Reduction methods, the nonlinearities are treated separately, resulting in two distinct sets of elements. In all these elements, the solution $(\tilde{\boldsymbol{u}}, \tilde{\mathrm{y}})$ has to be computed based on the reduced solution $(\tilde{\boldsymbol{u}}^r, \tilde{\mathrm{y}}^r)$ with $N^{\boldsymbol{u}}_{\mathrm{eval}}$ and $N^{\mathrm{y}}_{\mathrm{eval}}$ denoting the number of DoFs involved in these operation performed in every iteration of the nonlinear solver.

The errors obtained by DEIM, Gappy POD and RID are in the same range, whilst the errors for EC are at least of one order of magnitude larger. The accuracy of Gappy POD is superior to DEIM as linear regression performs better than interpolation and additionally increases the robustness. For DEIM, 18 out of 2000 runs failed to converge, whereas no such deficiencies are observed for the other methods.

The computation times were measured on a single core (AMD™ Ryzen™ 1950X CPU @4 GHz) without using any kind of parallelisation. The speed-up is defined as the ratio of time needed to solve the FOM and the ROM for the 2000 parameters in $\mathcal{V}_{\mathrm{II}}$, which does not include the time to calculate the homogenised quantities. The speed-up for DEIM is the greatest as the least number of solution and nonlinearity evaluations had to be performed, with Gappy POD being second due to more evaluations. The EC and RID reduced models are considerably slower as both methods need to evaluate the nonlinearities for a larger number of elements to gain comparable accuracy.

Table 4. Comparison of selected hyper-reduced models using a reduced basis with $n_{\boldsymbol{u}} = 20$ and $n_{\mathrm{y}} = 10$.

	DEIM	**Gappy POD**	**EC**	**RID**
$m_{\boldsymbol{u}}$ & $m_{\mathrm{y}} \rightarrow m$	50 & 33 $\rightarrow$ 71	69 & 75 $\rightarrow$ 112	120 & 110 $\rightarrow$ 219	129 & 72 $\rightarrow$ 183
$N^{\boldsymbol{u}}_{\mathrm{eval}}$ & $N^{\mathrm{y}}_{\mathrm{eval}}$	1000 & 500	1468 & 734	3400 & 1700	1938 & 969
$\mathrm{Err}_{\overline{\boldsymbol{P}}}(\mathcal{V}_{\mathrm{II}})$	1.14×10^{-3}	3.86×10^{-4}	1.85×10^{-2}	5.99×10^{-4}
$\mathrm{Err}_{\overline{\mathbb{B}}}(\mathcal{V}_{\mathrm{II}})$	4.71×10^{-4}	2.78×10^{-4}	4.63×10^{-3}	5.18×10^{-4}
n_{fail}	18	0	0	0
speed-up	208	131	23	32

Table 5. Parameters of the hyper-reduced models used in Table 4.

DEIM	$r_{\boldsymbol{u}} = 37$ and $r_{\mathrm{y}} = 25$
Gappy POD	$r_{\boldsymbol{u}} = 36$ and $r_{\mathrm{y}} = 24$, $p_{\boldsymbol{u}} = 50$ and $p_{\mathrm{y}} = 58$
EC	$m_{\boldsymbol{u}} = 120$ and $m_{\mathrm{y}} = 110$
RID	$l = 1$

6. Conclusions

In this work, we applied the tools of reduced-order modelling to the problems arising in computational homogenisation in magneto-mechanics. The main focus was the investigation and comparison of different hyper-reduction techniques with respect to accuracy and robustness. Collateral basis methods like DEIM and Gappy POD are the fastest and most accurate, but are susceptible to robustness deficiencies. This is particularly true for DEIM, for which we could not build sufficiently robust models in the course of this work. Gappy POD diminishes that issue by using linear regression, providing adequately robust models. Additionally, unlike Gappy POD, DEIM does not offer the option to evaluate the mechanical and magnetic nonlinearity in the same elements, resulting in more expensive reduced models. For those reasons, DEIM will not be considered in future studies. A disadvantage shared among the collateral basis methods is that instances of the FE residuum have to be collected from non-equilibrated states, which results in more expensive POD computations.

EC and RID solve the weak form in a subdomain, to which the first refers to reduced mesh and the latter to reduced integration domain. To obtain similar accuracy as collateral basis methods, EC

and RID have to perform more function evaluations and are therefore more expensive. However, their robustness is superior to the collateral basis methods and hence they are particularly suitable for multi-query frameworks like the FE^2 method. For problems with stronger nonlinearities, e.g. due to complex material models, rate-dependence, plasticity and many more, the robustness superiority will be even more pronounced.

The next step is to equip the reduced models with an auxiliary basis to efficiently compute the homogenised Piola stress and magnetic induction, which can be utilised in a perturbation-type method [50] to obtain the macroscopic tangent moduli. Similarly, the macroscopic tangent moduli could be computed adapting the method described in [51] for the reduced model.

Author Contributions: Conceptualization, B.B.; Investigation, B.B.; Software, B.B., D.D.; Supervision, P.S.; Writing—Original Draft, B.B.; Writing—Review and Editing, D.D., J.M. and P.S.

Funding: Benjamin Brands was partially supported by the Indo-German exchange programm "Multiscale Modeling, Simulation and Optimization for Energy, Advanced Materials and Manufacturing" (MMSO) funded by DAAD (Germany) and UGC (India). Denis Davydov was supported by the German Research Foundation (DFG), grant DA 1664/2-1.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MRE	Magneto-Rheological Elastomer
BVP	Boundary Value Problem
RVE	Representative Volume Element
ROM	Reduced-Order Model
FEM	Finite Element Method
DoF	Degree of Freedom
FOM	Full-Order Model
pPDE	parametrised Partial Differential Equation
POD	Proper Orthogonal Decomposition
SVD	Singular Value Decomposition
EIM	Empirical Interpolation Method
DEIM	Discrete Empirical Interpolation Method
GNAT	Gauss–Newton with Approximated Tensors
EC	Empirical Cubature
ECSW	Energy-Conserving Sampling and Weighting
RID	Reduced Integration Domain

Appendix A. DEIM

The classical DEIM Algorithm 1 [19] determines the interpolation indices iteratively. In iteration i, the index ρ_i to be added is the entry where the approximation of the basis vector $\boldsymbol{\phi}_i$ by the preceding vectors $\left\{\boldsymbol{\phi}_j\right\}_{j=1}^{i}$ exhibits the largest error. The basis vectors $\left\{\boldsymbol{\phi}_i\right\}_{i=1}^{r}$ have to be linearly independent, which is guaranteed by using a basis computed by POD.

Algorithm 1: DEIM Algorithm

Input: $\{\boldsymbol{\phi}_i\}_{i=1}^{r} \subset \mathbb{R}^{\mathrm{N}}$
Output: $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_r] \in \mathbb{N}^r$, $\mathbf{P}$
$\rho_1 = \underset{a=1,\ldots,\mathrm{N}}{\arg\max} \left|\boldsymbol{\phi}_1[a]\right|$
$S = [\boldsymbol{\phi}_1]$, $\mathbf{P} = \left[\boldsymbol{e}_{\rho_1}\right]$, $\boldsymbol{\rho} = [\rho_1]$
for i=2 **to** r **do**
  Solve for $\mathbf{c}$ in $\left(\mathbf{P}^\top S\right)\mathbf{c} = \mathbf{P}^\top \boldsymbol{\phi}_i$
  $\boldsymbol{r} = \boldsymbol{\phi}_i - S\boldsymbol{c}$
  $\rho_i = \underset{a=1,\ldots,\mathrm{N}}{\arg\max} \left|\boldsymbol{\phi}_i[a]\right|$
  $\boldsymbol{\rho} \leftarrow [\boldsymbol{\rho}, \rho_i]$, $\mathbf{P} \leftarrow \left[\mathbf{P}, \boldsymbol{e}_{\rho_i}\right]$, $S \leftarrow [S, \boldsymbol{\phi}_i]$
end

Appendix B. Gappy POD and GNAT

Like the DEIM Algorithm 1, the adapted point search Algorithm 2 [21] seeks to minimise the error in approximating the bases $\left\{\left\{\boldsymbol{\phi}_i^c\right\}_{i=1}^{r_c}\right\}_{c=1}^{n_{\text{components}}}$ and chooses the gappy points accordingly. The algorithm processes nodes instead of indices and is therefore well-suited for vector-valued or multi-physic problems. Differences in scale are taken care of by using normalised maxima for the different fields.

Algorithm 2: Greedy Point Search

Input: reduced bases $\left\{ \left\{ \boldsymbol{\phi}_i^c \right\}_{i=1}^{r_c} \right\}_{c=1}^{n_{\text{components}}}$, number of sampling points p

Output: $\{\mathbf{P}_c\}_{c=1}^{n_{\text{components}}}$

number of greedy iterations: $g = \min\left(\left\{ r_1, \ldots, r_{n_{\text{components}}}, \mathrm{p} \right\} \right)$

number of sampling points at iteration i: $s(i)$

$$s(i) = \left\lfloor \frac{w \cdot \mathrm{p}}{g} \right\rfloor \text{ with } w = \left\lceil \frac{g}{\mathrm{p}} \right\rceil;\ \text{if } w = 1 \text{ and } i \leq (\mathrm{p} \bmod g) :\ s(i) \leftarrow s(i) + 1$$

number of basis vectors to be added/processed at iteration i:

$$q_c(i) = \left\lfloor \frac{r_c}{g} \right\rfloor;\ \text{if } i \leq (r_c \bmod g) :\ q_c(i) \leftarrow q_c(i) + 1 \text{ for } c = 1, \ldots, n_{\text{components}}$$

for *c=1* **to** $n_{\text{components}}$ **do**

  $\left[\boldsymbol{S}_1^c, \ldots, \boldsymbol{S}_{q_c(1)}^c \right] \leftarrow \left[\boldsymbol{\phi}_1^c, \ldots, \boldsymbol{\phi}_{q_c(1)}^c \right]$ `// vectors to be processed in first iteration`

end

for *i=1* **to** g **do**

  for *j=1* **to** $s(i)$ **do**

    for *c=1* **to** $n_{\text{components}}$ **do**

      $n_{\max}^c \leftarrow \underset{l \in \{1,\ldots,n_{\text{points}}\}}{\arg\max} \sum_{q=1}^{q_c(i)} \left\| \mathbf{S}_q^c[l] \right\|^2$ `// location of component maximum`

    end

    $n \leftarrow \underset{l \in \{1,\ldots,n_{\text{points}}\} \setminus \mathcal{N}}{\arg\max} \sum_{c=1}^{n_{\text{components}}} \sum_{q=1}^{q_c(i)} \frac{\left\| \mathbf{S}_q^c[l] \right\|^2}{\left\| \mathbf{S}_q^c[n_{\max}^c] \right\|^2}$ `// location of combined maximum`

    for *c=1* **to** $n_{\text{components}}$ **do**

      $I_c = \mathrm{DoFs}_c(n)$ `// get set of component DoFs attached to point`

      $\mathbf{P}_c \leftarrow \left[\mathbf{P}_c, \left[\mathbf{e}_{I_c[1]}, \ldots, \mathbf{e}_{I_c[|I_c|]} \right] \right]$

    end

    $\mathcal{N} \leftarrow \mathcal{N} + n$ `// update set of selected points`

  end

  for *c=1* **to** $n_{\text{components}}$ **do**

    for *j=1* **to** $q_c(i)$ **do**

      $\tilde{\boldsymbol{\phi}}_{Q_c+j}^c = \underset{\mathbf{a}}{\arg\min} \left\| \mathbf{P}_c^\top \left[\boldsymbol{\phi}_1^c, \ldots, \boldsymbol{\phi}_{Q_c}^c \right] \mathbf{a} - \mathbf{P}_c^\top \boldsymbol{\phi}_{Q_c+j}^c \right\|_2^2$

      $\mathbf{S}_j^c \leftarrow \boldsymbol{\phi}_{Q_c+j}^c - \left[\boldsymbol{\phi}_1^c, \ldots, \boldsymbol{\phi}_{Q_c}^c \right] \tilde{\boldsymbol{\phi}}_{Q_c+j}^c$ `// vectors processed in next iteration`

    end

    $Q_c \leftarrow Q_c + q_c(i)$

  end

end

Appendix C. Empirical Cubature

To determine the reduced meshes and the weights, the minimisation problems based on (39)

$$(\boldsymbol{\omega}^{\boldsymbol{u}}, \mathcal{E}_{\boldsymbol{u}}) = \underset{\substack{\boldsymbol{w}\in\mathbb{R}_{+}^{\mathrm{m}_{\boldsymbol{u}}} \\ \mathcal{E}\subset\mathcal{T}}}{\arg\min} \sqrt{\sum_{j=1}^{\mathrm{n}_P}\sum_{i=1}^{\mathrm{n}_{\boldsymbol{u}}} \left({}^{\boldsymbol{u}}\check{e}_{ij}(\boldsymbol{w},\mathcal{E})\right)^2} \quad \text{and} \quad (\boldsymbol{\omega}^{\mathbb{y}}, \mathcal{E}_{\mathbb{y}}) = \underset{\substack{\boldsymbol{w}\in\mathbb{R}_{+}^{\mathrm{m}_{\mathbb{y}}} \\ \mathcal{E}\subset\mathcal{T}}}{\arg\min} \sqrt{\sum_{j=1}^{\mathrm{n}_{\mathbb{B}}}\sum_{i=1}^{\mathrm{n}_{\mathbb{y}}} \left({}^{\mathbb{y}}\check{e}_{ij}(\boldsymbol{w},\mathcal{E})\right)^2} \tag{A1}$$

have to be solved and read in matrix format as

$$(\boldsymbol{\omega}^{\boldsymbol{u}}, \mathcal{E}_{\boldsymbol{u}}) = \underset{\substack{\boldsymbol{w}\in\mathbb{R}_{+}^{\mathbf{m}_{\boldsymbol{u}}} \\ \mathcal{E}\in\mathcal{T}}}{\arg\min} \left\| \check{\boldsymbol{J}}^{\boldsymbol{u}}_{\mathcal{E}}\boldsymbol{w} - \check{\boldsymbol{b}}^{\boldsymbol{u}} \right\|_2 \quad \text{and} \quad (\boldsymbol{\omega}^{\mathbb{y}}, \mathcal{E}_{\mathbb{y}}) = \underset{\substack{\boldsymbol{\omega}\in\mathbb{R}_{+}^{\mathbf{m}_{\mathbb{y}}} \\ \mathcal{E}\in\mathcal{T}}}{\arg\min} \left\| \check{\boldsymbol{J}}^{\mathbb{y}}_{\mathcal{E}}\boldsymbol{w} - \check{\boldsymbol{b}}^{\mathbb{y}} \right\|_2, \tag{A2}$$

$$\text{with } \check{\boldsymbol{J}}^{\boldsymbol{u}} \in \mathbb{R}^{\mathrm{n}_{\boldsymbol{u}}\mathrm{n}_P\times \mathrm{M}}, \check{\boldsymbol{J}}^{\mathbb{y}} \in \mathbb{R}^{\mathrm{n}_{\mathbb{y}}\mathrm{n}_{\mathbb{B}}\times \mathrm{M}}, \check{\boldsymbol{b}}^{\boldsymbol{u}} \in \mathbb{R}^{\mathrm{n}_{\boldsymbol{u}}\mathrm{n}_P} \text{ and } \check{\boldsymbol{b}}^{\mathbb{y}} \in \mathbb{R}^{\mathrm{n}_{\mathbb{y}}\mathrm{n}_{\mathbb{B}}},$$

$$\check{\boldsymbol{J}}^{\boldsymbol{u}} = \begin{bmatrix} \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_1 : \boldsymbol{\phi}^P_1 \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_1 : \boldsymbol{\phi}^P_1 \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_{\mathrm{n}_{\boldsymbol{u}}} : \boldsymbol{\phi}^P_1 \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_{\mathrm{n}_{\boldsymbol{u}}} : \boldsymbol{\phi}^P_1 \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_{\mathrm{n}_{\boldsymbol{u}}} : \boldsymbol{\phi}^P_{\mathrm{n}_P} \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_{\mathrm{n}_{\boldsymbol{u}}} : \boldsymbol{\phi}^P_{\mathrm{n}_P} \,\mathrm{d}V \end{bmatrix} \quad \check{\boldsymbol{b}}^{\boldsymbol{u}} = \sum_{e=1}^{\mathrm{M}} \begin{bmatrix} \int\limits_{\Omega_e} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_1 : \boldsymbol{\phi}^P_1 \,\mathrm{d}V \\ \vdots \\ \int\limits_{\Omega_e} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_{\mathrm{n}_{\boldsymbol{u}}} : \boldsymbol{\phi}^P_1 \,\mathrm{d}V \\ \vdots \\ \int\limits_{\Omega_e} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}^{\boldsymbol{u}}_{\mathrm{n}_{\boldsymbol{u}}} : \boldsymbol{\phi}^P_{\mathrm{n}_P} \,\mathrm{d}V \end{bmatrix}$$

$$\check{\boldsymbol{J}}^{\mathbb{y}} = \begin{bmatrix} \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_1 \cdot \boldsymbol{\phi}^{\mathbb{B}}_1 \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_1 \cdot \boldsymbol{\phi}^{\mathbb{B}}_1 \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_{\mathrm{n}_{\mathbb{y}}} \cdot \boldsymbol{\phi}^{\mathbb{B}}_1 \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_{\mathrm{n}_{\mathbb{y}}} \cdot \boldsymbol{\phi}^{\mathbb{B}}_1 \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_{\mathrm{n}_{\mathbb{y}}} \cdot \boldsymbol{\phi}^{\mathbb{B}}_{\mathrm{n}_{\mathbb{B}}} \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_{\mathrm{n}_{\mathbb{y}}} \cdot \boldsymbol{\phi}^{\mathbb{B}}_{\mathrm{n}_{\mathbb{B}}} \,\mathrm{d}V \end{bmatrix} \quad \check{\boldsymbol{b}}^{\mathbb{y}} = \sum_{e=1}^{\mathrm{M}} \begin{bmatrix} \int\limits_{\Omega_e} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_1 \cdot \boldsymbol{\phi}^{\mathbb{B}}_1 \,\mathrm{d}V \\ \vdots \\ \int\limits_{\Omega_e} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_{\mathrm{n}_{\mathbb{y}}} \cdot \boldsymbol{\phi}^{\mathbb{B}}_1 \,\mathrm{d}V \\ \vdots \\ \int\limits_{\Omega_e} \nabla_{\boldsymbol{X}}\varphi^{\mathbb{y}}_{\mathrm{n}_{\mathbb{y}}} \cdot \boldsymbol{\phi}^{\mathbb{B}}_{\mathrm{n}_{\mathbb{B}}} \,\mathrm{d}V \end{bmatrix}.$$

Note that the problems in (A2) allow for the trivial solutions $\boldsymbol{\omega}^{\boldsymbol{u}} = \mathbf{0}$ and $\boldsymbol{\omega}^{\boldsymbol{u}} = \mathbf{0}$. Any POD mode $\hat{\boldsymbol{\phi}}^P_j$ or $\hat{\boldsymbol{\phi}}^{\mathbb{B}}_j$ is a linear combination of the snapshots $\left\{\hat{\boldsymbol{P}}_1, \ldots, \hat{\boldsymbol{P}}_{\mathrm{n}_{\mathrm{s}}}\right\}$ or $\left\{\hat{\mathbb{B}}_1, \ldots, \hat{\mathbb{B}}_{\mathrm{n}_{\mathrm{s}}}\right\}$ and, as the snapshots are taken from states of equilibrium, the right-hand sides become $\check{\boldsymbol{b}}^{\boldsymbol{u}} = \mathbf{0}$ and $\check{\boldsymbol{b}}^{\mathbb{y}} = \mathbf{0}$. Therefore, problems (A2) are regularised by adding the constraints $\sum\limits_{e\in\mathcal{E}_{\boldsymbol{u}}} \omega^{\boldsymbol{u}}_e = V$ and $\sum\limits_{e\in\mathcal{E}_{\mathbb{y}}} \omega^{\mathbb{y}}_e = V$. By subtracting the volume averaged row-sums, the regularised minimisation problems (A3) are obtained and approximately solved using Algorithm 3:

$$(\boldsymbol{\omega}^{\boldsymbol{u}}, \mathcal{E}_{\boldsymbol{u}}) = \underset{\substack{\boldsymbol{w}\in\mathbb{R}_{+}^{\mathbf{m}_{\boldsymbol{u}}} \\ \mathcal{E}\in\mathcal{T}}}{\arg\min} \left\| \boldsymbol{J}_{\mathcal{E}}^{\boldsymbol{u}}\boldsymbol{w} - \boldsymbol{b}^{\boldsymbol{u}} \right\|_2 \quad \text{and} \quad (\boldsymbol{\omega}^{\mathbb{y}}, \mathcal{E}_{\mathbb{y}}) = \underset{\substack{\boldsymbol{\omega}\in\mathbb{R}_{+}^{\mathbf{m}_{\mathbb{y}}} \\ \mathcal{E}\in\mathcal{T}}}{\arg\min} \left\| \boldsymbol{J}_{\mathcal{E}}^{\mathbb{y}}\boldsymbol{w} - \boldsymbol{b}^{\mathbb{y}} \right\|_2, \tag{A3}$$

with $\boldsymbol{J}^{\boldsymbol{u}} = \begin{bmatrix} \bar{\boldsymbol{J}}^{\boldsymbol{u}} \\ \mathbf{1}^\top \end{bmatrix} \in \mathbb{R}^{(\mathrm{n}_{\boldsymbol{u}}\mathrm{n}_P+1)\times \mathrm{M}}$, $\boldsymbol{J}^{\mathbb{y}} = \begin{bmatrix} \bar{\boldsymbol{J}}^{\mathbb{y}} \\ \mathbf{1}^\top \end{bmatrix} \in \mathbb{R}^{(\mathrm{n}_{\mathbb{y}}\mathrm{n}_{\mathbb{B}}+1)\times \mathrm{M}}$, $\boldsymbol{b}^{\boldsymbol{u}} = \begin{bmatrix} \mathbf{0} \\ V \end{bmatrix} \in \mathbb{R}^{\mathrm{n}_{\boldsymbol{u}}\mathrm{n}_P+1}$

and $\boldsymbol{b}^{\mathbb{y}} = \begin{bmatrix} \mathbf{0} \\ V \end{bmatrix} \in \mathbb{R}^{\mathrm{n}_{\mathbb{y}}\mathrm{n}_{\mathbb{B}}+1}$

$$\bar{\boldsymbol{J}}^{\boldsymbol{u}} = \begin{bmatrix} \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_1^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_1^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_1^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_1^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_1^P \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_{\mathrm{n}_P}^P \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_{\mathrm{n}_P}^P \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_{\mathrm{n}_P}^P \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}_{\mathrm{n}_{\boldsymbol{u}}}^{\boldsymbol{u}} : \boldsymbol{\phi}_{\mathrm{n}_P}^P \,\mathrm{d}V \end{bmatrix},$$

$$\bar{\boldsymbol{J}}^{\mathbb{y}} = \begin{bmatrix} \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\varphi_1^{\mathbb{y}} : \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\varphi_1^{\mathbb{y}} : \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\varphi_1^{\mathbb{y}} \cdot \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\varphi_1^{\mathbb{y}} : \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} \cdot \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} : \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} \cdot \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} : \boldsymbol{\phi}_1^{\mathbb{B}} \,\mathrm{d}V \\ \vdots & \ddots & \vdots \\ \int\limits_{\Omega_1} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} \cdot \boldsymbol{\phi}_{\mathrm{n}_{\mathbb{B}}}^{\mathbb{B}} \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} : \boldsymbol{\phi}_{\mathrm{n}_{\mathbb{B}}}^{\mathbb{B}} \,\mathrm{d}V & \cdots & \int\limits_{\Omega_{\mathrm{M}}} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} \cdot \boldsymbol{\phi}_{\mathrm{n}_{\mathbb{B}}}^{\mathbb{B}} \,\mathrm{d}V - \frac{1}{V}\int\limits_{\Omega} \nabla_{\boldsymbol{X}}\varphi_{\mathrm{n}_{\mathbb{y}}}^{\mathbb{y}} : \boldsymbol{\phi}_{\mathrm{n}_{\mathbb{B}}}^{\mathbb{B}} \,\mathrm{d}V \end{bmatrix}.$$

Algorithm 3: Greedy Mesh Sampling

Input: $\boldsymbol{J}$, tol, m
Output: $\boldsymbol{\omega}$, E
initialisation: $E \leftarrow \emptyset$, set of candidates $C \leftarrow \{1, \ldots, \mathrm{M}\}$, $\boldsymbol{r} \leftarrow \boldsymbol{b}$
do
  $e = \underset{\tilde{e}\in C}{\arg\max} \left\langle \boldsymbol{J}[\tilde{e}]/\left\|\boldsymbol{J}[\tilde{e}]\right\|_2, \boldsymbol{r}/\|\boldsymbol{r}\|_2 \right\rangle$ `// determine new element`
  $E \leftarrow E \cup e$, $C \leftarrow C \setminus e$
  build $\boldsymbol{J}_E$ from columns of $\boldsymbol{J}$ based on E
  $\boldsymbol{\omega} = \underset{\boldsymbol{w}\in\mathbb{R}^{|E|}}{\arg\min} \|\boldsymbol{J}_E\boldsymbol{w} - \boldsymbol{b}\|_2^2$ `// solve least-squares`
  if $\boldsymbol{\omega}[i] < 0$ *for* $i = 1, \ldots, |E|$ **then**
    $\boldsymbol{\omega} = \underset{\boldsymbol{w}\in\mathbb{R}_{+}^{|E|}}{\arg\min} \|\boldsymbol{J}_E\boldsymbol{w} - \boldsymbol{b}\|_2^2$ `// solve non-negative least-squares`
    $E \leftarrow E \setminus E_0$ with $E_0 = \{e \in E : \boldsymbol{\omega}[e] = 0\}$
    $C \leftarrow C \cup E_0$
    $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega}(E)$
  end
  $\boldsymbol{r} \leftarrow \boldsymbol{b} - \boldsymbol{J}_E\boldsymbol{\omega}$ `// update residual`
while $\dfrac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|} > tol \ \wedge \ |E| < m$

Appendix D. Reduced Integration Domain

Algorithm 4 constructs the reduced integration domain for a basis $\{\boldsymbol{\phi}_i\}_{i=1}^{r}$. The application of the DEIM Algorithm 1 helps to identify areas of interest, e.g. where the basis exhibits significant gradients. At first, all elements containing DEIM indices form the reduced integration domain. Hereafter, the elements in the surrounding layers can be included for accuracy reasons. It is noteworthy that Algorithm 2 or alternatives can be used to determine the initial reduced domain.

Algorithm 4: Determining Reduced Integration Domain

Input: POD basis $\{\boldsymbol{\phi}_i\}_{i=1}^{r} \subset \mathbb{R}^{\mathrm{N}}$, number of neighbouring element layers l
Output: $\mathbf{P}_{\mathrm{RID}}$, Ω_{RID}

get DEIM indices $\boldsymbol{\rho} \in \mathbb{N}^r$ by applying Algorithm 1 to $\{\boldsymbol{\phi}_i\}_{i=1}^{r}$
$\Omega_{\mathrm{RID}} \leftarrow$ ContainingElements $(\boldsymbol{\rho})$ `// collect elements containing the DEIM indices`
for *i=1* **to** l **do**
  $\Omega_{\mathrm{RID}} \leftarrow \Omega_{\mathrm{RID}} \cup$ NeighbouringElements (Ω_{RID}) `// add neighbouring elements`
end
setup $\mathbf{P}_{\mathrm{RID}}$ based on interior DoFs (primary) in Ω_{RID}

References

1. Feyel, F.; Chaboche, J.L. FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Comput. Methods Appl. Mech. Eng.* **2000**, *183*, 309–330.
2. Saeb, S.; Steinmann, P.; Javili, A. Aspects of computational homogenization at finite deformations: A unifying review from Reuss' to Voigt's bound. *Appl. Mech. Rev.* **2016**, *68*, 050801.
3. Bartuschat, D.; Gmeiner, B.; Thoennes, D.; Kohl, N.; Rüde, U.; Drzisga, D.; Huber, M.; John, L.; Waluga, C.; Wohlmuth, B.I.; et al. A Finite Element Multigrid Framework for Extreme-Scale Earth Mantle Convection Simulations. In Proceedings of the SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP 18), Tokyo, Japan, 7–10 March 2018.
4. Michel, J.C.; Suquet, P. A model-reduction approach in micromechanics of materials preserving the variational structure of constitutive relations. *J. Mech. Phys. Solids* **2016**, *90*, 254–285.
5. Fritzen, F.; Leuschner, M. Reduced basis hybrid computational homogenization based on a mixed incremental formulation. *Comput. Methods Appl. Mech. Eng.* **2013**, *260*, 143–154.
6. Chinesta, F.; Ammar, A.; Leygue, A.; Keunings, R. An overview of the proper generalized decomposition with applications in computational rheology. *J. Non-Newton. Fluid Mech.* **2011**, *166*, 578–592.
7. Cremonesi, M.; Néron, D.; Guidault, P.A.; Ladevèze, P. A PGD-based homogenization technique for the resolution of nonlinear multiscale problems. *Comput. Methods Appl. Mech. Eng.* **2013**, *267*, 275–292.
8. Sirovich, L. Turbulence and the Dynamics of Coherent Structures Part I: Coherent Structures. *Q. Appl. Math.* **1987**, *45*, 561–571.
9. Himpe, C.; Leibner, T.; Rave, S. Hierarchical Approximate Proper Orthogonal Decomposition. *SIAM J. Sci. Comput.* **2018**, *40*, A3267–A3292.
10. Paul-Dubois-Taine, A.; Amsallem, D. An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models. *Int. J. Numer. Methods Eng.* **2014**, *102*, 1262–1292.
11. Brands, B.; Mergheim, J.; Steinmann, P. Reduced-order modelling for linear heat conduction with parametrised moving heat sources. *GAMM-Mitteilungen* **2016**, *39*, 170–188.
12. Hesthaven, J.S.; Rozza, G.; Stamm, B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*; Springer: Cham, Switzerland, 2017.
13. Quarteroni, A.; Manzoni, A.; Negri, F. *Reduced Basis Methods for Partial Differential Equations: An Introduction*; Springer: Cham, Switzerland, 2015; Volume 92.
14. Rozza, G.; Huynh, D.B.P.; Patera, A.T. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.* **2008**, *15*, 229–275.

15. Grepl, M.A.; Patera, A.T. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM Math. Model. Numer. Anal.* **2005**, *39*, 157–181.
16. Yvonnet, J.; He, Q.C. The reduced model multiscale method (R3M) for the nonlinear homogenization of hyperelastic media at finite strains. *J. Comput. Phys.* **2007**, *223*, 341–368.
17. Radermacher, A.; Bednarcyk, B.A.; Stier, B.; Simon, J.; Zhou, L.; Reese, S. Displacement-based multiscale modeling of fiber-reinforced composites by means of proper orthogonal decomposition. *Adv. Model. Simul. Eng. Sci.* **2016**, *3*, 29.
18. Grepl, M.A.; Maday, Y.; Nguyen, N.C.; Patera, A.T. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM M2AN* **2007**, *41*, 575–605.
19. Chaturantabut, S.; Sorensen, D.C. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764.
20. Everson, R.; Sirovich, L. Karhunen-Loeve procedure for gappy data. *J. Opt. Soc. Am. A* **1995**, *12*, 1657–1664.
21. Carlberg, K.; Farhat, C.; Cortial, J.; Amsallem, D. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.* **2013**, *242*, 623–647.
22. Goury, O.; Amsallem, D.; Bordas, S.P.A.; Liu, W.K.; Kerfriden, P. Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: From dissipation-driven random selection to Bayesian optimization. *Comput. Mech.* **2016**, *58*, 213–234.
23. Ghavamian, F.; Tiso, P.; Simone, A. POD–DEIM model order reduction for strain-softening viscoplasticity. *Comput. Methods Appl. Mech. Eng.* **2017**, *317*, 458–479.
24. Tiso, P.; Rixen, D.J. Discrete Empirical Interpolation Method for Finite Element Structural Dynamics. In *Topics in Nonlinear Dynamics, Volume 1: Proceedings of the 31st IMAC, A Conference on Structural Dynamics, Garden Grove, CA, USA, 11–14 February 2013*; Kerschen, G., Adams, D., Carrella, A., Eds.; Springer: New York, NY, USA, 2013; pp. 203–212.
25. Hernández, J.; Oliver, J.; Huespe, A.; Caicedo, M.; Cante, J. High-performance model reduction techniques in computational multiscale homogenization. *Comput. Methods Appl. Mech. Eng.* **2014**, *276*, 149–189.
26. Soldner, D.; Brands, B.; Zabihyan, R.; Steinmann, P.; Mergheim, J. A numerical study of different projection-based model reduction techniques applied to computational homogenisation. *Comput. Mech.* **2017**, *60*, 613–625.
27. An, S.S.; Kim, T.; James, D.L. Optimizing Cubature for Efficient Integration of Subspace Deformations. *ACM Trans. Graph.* **2008**, *27*, 165.
28. Farhat, C.; Avery, P.; Chapman, T.; Cortial, J. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *Int. J. Numer. Methods Eng.* **2014**, *98*, 625–662.
29. Hernández, J.; Caicedo, M.; Ferrer, A. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Comput. Methods Appl. Mech. Eng.* **2017**, *313*, 687–722.
30. van Tuijl, R.A.; Remmers, J.J.C.; Geers, M.G.D. Integration efficiency for model reduction in micro-mechanical analyses. *Comput. Mech.* **2017**, *62*, 151–169.
31. Ryckelynck, D. A priori hyperreduction method: An adaptive approach. *J. Comput. Phys.* **2005**, *202*, 346–366.
32. Ryckelynck, D. Hyper-reduction of mechanical models involving internal variables. *Int. J. Numer. Methods Eng.* **2009**, *77*, 75–89.
33. Ryckelynck, D.; Benziane, D.M. Multi-level A Priori Hyper-Reduction of mechanical models involving internal variables. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 1134–1142.
34. Ryckelynck, D.; Lampoh, K.; Quilicy, S. Hyper-reduced predictions for lifetime assessment of elasto-plastic structures. *Meccanica* **2016**, *51*, 309–317.
35. Fritzen, F.; Haasdonk, B.; Ryckelynck, D.; Schöps, S. An Algorithmic Comparison of the Hyper-Reduction and the Discrete Empirical Interpolation Method for a Nonlinear Thermal Problem. *Math. Comput. Appl.* **2018**, *23*, 8.
36. Astrid, P.; Weiland, S.; Willcox, K.; Backx, T. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Trans. Autom. Control* **2008**, *53*, 2237–2251.
37. Dimitriu, G.; Ştefănescu, R.; Navon, I.M. Comparative numerical analysis using reduced-order modeling strategies for nonlinear large-scale systems. *J. Comput. Appl. Math.* **2017**, *310*, 32–43.

38. Walter, B.L.; Pelteret, J.P.; Kaschta, J.; Schubert, D.W.; Steinmann, P. Preparation of magnetorheological elastomers and their slip-free characterization by means of parallel-plate rotational rheometry. *Smart Mater. Struct.* **2017**, *26*, 085004.
39. Chatzigeorgiou, G.; Javili, A.; Steinmann, P. Unified magnetomechanical homogenization framework with application to magnetorheological elastomers. *Math. Mech. Solids* **2014**, *19*, 193–211.
40. Zabihyan, R.; Mergheim, J.; Javili, A.; Steinmann, P. Aspects of computational homogenization in magneto-mechanics: Boundary conditions, RVE size and microstructure composition. *Int. J. Solids Struct.* **2018**, *130*, 105–121.
41. Javili, A.; Chatzigeorgiou, G.; Steinmann, P. Computational homogenization in magneto-mechanics. *Int. J. Solids Struct.* **2013**, *50*, 4197–4216.
42. Holmes, P.; Lumley, J.L.; Berkooz, G.; Rowley, C.W. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*; Cambridge University Press: Cambridge, UK, 2012.
43. Volkwein, S. *Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling*; Lecture Notes; University of Konstanz: Konstanz, Germany, 2013; Volume 4.
44. Chaturantabut, S.; Sorensen, D. A State Space Error Estimate for POD-DEIM Nonlinear Model Reduction. *SIAM J. Numer. Anal.* **2012**, *50*, 46–63.
45. Wirtz, D.; Sorensen, D.; Haasdonk, B. A Posteriori Error Estimation for DEIM Reduced Nonlinear Dynamical Systems. *SIAM J. Sci. Comput.* **2014**, *36*, A311–A338.
46. Chapman, T.; Avery, P.; Collins, P.; Farhat, C. Accelerated mesh sampling for the hyper reduction of nonlinear computational models. *Int. J. Numer. Methods Eng.* **2017**, *109*, 1623–1654.
47. Alzetta, G.; Arndt, D.; Bangerth, W.; Boddu, V.; Brands, B.; Davydov, D.; Gassmöller, R.; Heister, T.; Heltai, L.; Kormann, K.; et al. The deal.II Library, Version 9.0. *J. Numer. Math.* **2018**, *26*, 173–183.
48. Stoyanov, M. Adaptive Sparse Grid Construction in a Context of Local Anisotropy and Multiple Hierarchical Parents. In *Sparse Grids and Applications—Miami 2016*; Garcke, J., Pflüger, D., Webster, C.G., Zhang, G., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 175–199.
49. Garcke, J.; Griebel, M. *Sparse Grids and Applications*, 1st ed.; Lecture Notes in Computational Science and Engineering, Springer Science & Business Media: Cham, Switzerland, 2012; Volume 88.
50. Miehe, C. Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. *Comput. Methods Appl. Mech. Eng.* **1996**, *134*, 223–240.
51. Pelteret, J.P.; Steinmann, P. *Magneto-Active Polymers: Fabrication, Characterisation, Modelling and Simulation at the Micro- and Macro-Scale*; de Gruyter Mouton: Berlin, Germany, 2019. In preparation.