Adv. Geosci., 68, 1–11, 2026
https://doi.org/10.5194/adgeo-68-1-2026
© Author(s) 2026. This work is distributed under
the Creative Commons Attribution 4.0 License.

Advances in Geosciences
Open Access
EGU

Platypus: a Python package for automated surrogate-assisted analysis of process-based simulations via the non-intrusive reduced-basis method

Jorge Nicolas Hayek[1], **Mauro Cacace**[1], **Marthe Faber**[1], **Yulia Gruzdeva**[1], **and Denise Degen**[2,1]

[1]GFZ Helmholtz-Zentrum für Geoforschung, Telegrafenberg, 14473 Potsdam, Germany
[2]Institute of Applied Geosciences, TU Darmstadt, Schnittspahnstraße 9, 64287 Darmstadt, Germany

Correspondence: Jorge Nicolas Hayek (jorge.nicolas.hayek.valencia@gfz.de)

Received: 19 June 2026 – Revised: 20 October 2026 – Accepted: 27 October 2026 – Published: 4 November 2026

Abstract. Modelling the dynamic response of reservoirs to fluid injection often relies on fully coupled thermo-hydro-mechanical models, which provide a high-fidelity representation of the governing physical processes. These models support operational and design decisions under significant geological and parametric uncertainties. However, their high computational cost severely limits their applicability in large-scale statistical analysis and thus limiting the potential to account for these uncertainties. We present `Platypus`, a Python package that bridges full-order process-based reservoir simulations and surrogate-based analysis. The package provides automated, reproducible workflows covering the full pipeline from parameter sampling and simulation interfacing, through ensemble curation and data management, to surrogate model construction and global sensitivity analysis. Surrogate models are built via the non-intrusive reduced-basis method to deliver computationally efficient, physics-consistent approximations of the transient reservoir response, enabling statistical analysis that require a large number of evaluations. The package is demonstrated on two geothermal applications: a synthetic reservoir benchmark and the Groß Schönebeck geothermal site. These examples illustrate the package's ability to handle time-dependent coupled hydrothermal and thermo-hydro-mechanical processes, manage large simulation ensembles, and perform global sensitivity analyses enabled by surrogates. `Platypus` is designed to be modular and extensible, providing a flexible tool for ensemble-based reservoir modelling workflows across a range of subsurface applications.

1 Introduction

Numerical models of reservoirs are essential tools to improve our understanding of subsurface conditions for real-world applications, yet they face significant computational challenges. The high-dimensional parameter space arising from the heterogeneity and uncertainty of the subsurface hinders a direct usage of high fidelity models for statistical analysis such as global sensitivity analysis, uncertainty quantification, model calibrations, and inversions, all of which require a large number of evaluations to adequately sample the parameter domain. Meeting these demands requires navigating a list of challenges including parameter sampling, data management, and model reduction. These problems have practical, existing solutions in isolation (e.g., McKay et al., 1979, for parameter sampling strategies; Sullivan and Kaszynski, 2019, for handling and curating simulation outputs; Hoyer and Hamman, 2017, and Abernathey et al., 2021, for large multidimensional datasets; and Degen et al., 2023, for surrogate modeling of physical processes in geoscientific applications), but collectively they represent a steep learning curve for a new user.

Surrogate modelling (also referred to as meta- or proxy modelling across communities) offers a computationally efficient alternative to full-order process-based simulations. Surrogate models broadly fall into three categories. Physics-based surrogate models, as constructed by, for instance, Hesthaven et al. (2016) and Benner et al. (2015), preserve the underlying physics, but are limited in the range of applications. So, problematic are general applications to hyperbolic and nonlinear problems, which constitute many geophysical applications such as geothermal, seismic and elec-

tromagnetic applications (Degen et al., 2023; Benner et al., 2015; Hesthaven and Ubbiali, 2018). Data-driven surrogate models, conversely, do not preserve the physics but are applicable even to complex nonlinear problems. Hybrid approaches aim to combine the strengths of both methods and overcome their individual limitations. Physics-informed neural networks (PINNs), for instance, introduce the physics as a constraint in the loss function, earning them the alternative name physics-constrained machine learning method. However, they do not guarantee internal consistency with physics for all states within the domain. The non-intrusive reduced-basis (NIRB) method (Degen et al., 2023; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018) is a model order reduction technique that constructs low-order approximations of high-fidelity models, aiming at preserving the input-output relationship of the models. Unlike classical machine learning techniques and PINNs, NIRB does not treat physics as data or as a constraint. Its computational efficiency makes it suitable for large statistical analysis such as probabilistic model calibrations, uncertainty quantification, and sensitivity analysis (Degen et al., 2022, 2023).

To assist in the steps of full-order model and surrogate model construction and application, we present `Platypus`[1], a software package that integrates sampling strategies, model curation and visualization, handling large distributed multidimensional ensemble data, surrogate training, and global sensitivity analysis into a single Python-based framework, bridging full-order process-based simulations and surrogate-based analysis. It provides automatic, reproducible pipelines and modular workflows adaptable to diverse scenarios and solvers. Example workflow scripts and tutorial Jupyter notebooks are included alongside the package.

2 Implementation

`Platypus` is an open-source Python package covering the full workflow from simulation interfacing to surrogate modeling, as illustrated in Fig. 1. Tailored to the finite element solver GOLEM/MOOSE (Jacquey and Cacace, 2017; Harbour et al., 2025) but being extendable to other simulation frameworks, it handles parameter sampling, input file generation, model curation, and ensemble assembly, all driven by a centralized configuration file. The package provides visualization and mesh handling through PyVista (Sullivan and Kaszynski, 2019), and stores simulation datasets in a labelled, distributed fashion using Zarr format with Xarray (Hoyer and Hamman, 2017), avoiding redundant storage. It further construct computationally efficient surrogate models via the non-intrusive reduced-basis method (Degen et al., 2023; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018), which can be applied directly to global sensitivity analysis through SALib (Iwanaga et al., 2022; Herman and Usher, 2017). Each of these steps is described in detail in the following subsections.

[1]The package is named after the creature that combines traits found separately elsewhere.

2.1 Parameter sampling

Sampling the parameter subspace for ensemble simulations involves a series of decisions. The number of parameters to vary, their ranges, and their distributions need to be specified. These choices are context-dependent, determined by the prior knowledge of the model setup, the workflow and the study objective. In iterative workflows involving a sensitivity analysis, the parameter set may shrink across iterations, as low influence parameters are identified and removed, requiring the parameter subspace to be redefined at each stage. Additionally, parameters may follow skewed or non-linear distributions (e.g., permeabilities are often defined in logarithmic scale while density or thermal conductivity use a linear scale), requiring sampling strategies that avoid bias in subsequent statistical analysis (e.g., Degen and Wellmann, 2026).

The configuration file contains the list of varying parameters, along with the respective value ranges, the ensemble size for training and validation realisation sets, and the sampling strategy for each set. `Platypus` supports the definition of arbitrary forward and inverse mappings to linearize each parameter distribution prior to sampling, and provides currently two sampling strategies: the Latin Hypercube sampling (LHS; Loh, 1996), and random sampling. The former is a sampling strategy widely adopted for uniform sampling of the parameter space to construct the training ensemble of models. The latter is often used for the validation ensemble.

Input files are generated from a reference input file template with placeholders for parameter values, ensuring that the ensemble members are built from a verified reference model. The resulting input files can be executed locally or on high-performance compute clusters.

2.2 Choice of simulator

The non-intrusive reduced-basis method is agnostic of the numerical method, making `Platypus` extensible to outputs from different process-based simulators, though it is currently tailored to GOLEM/MOOSE. GOLEM (Jacquey and Cacace, 2017) is an open-source finite element simulator built on the general Multiphysics Object-Oriented Simulation Environment (MOOSE; Harbour et al., 2025). It implements the coupled thermo-hydro-mechanical formulation describing fluid flow, heat transport, and mechanical deformation in porous media. The coupled system of equations is summarized below.

The governing equation for pore fluid pressure comes from the fluid mass balance, under Boussinesq approximation, as

$$\frac{1}{M_{\mathrm{B}}}\frac{\partial p_{\mathrm{f}}}{\partial t} - \beta_{\mathrm{b}}\frac{\partial T}{\partial t} + \alpha\dot{\underline{\underline{\boldsymbol{\epsilon}}}}_{kk} + (1-\alpha)\dot{\underline{\underline{\boldsymbol{\epsilon}}}}^{\star}_{kk} + \nabla\cdot\boldsymbol{q}_{\mathrm{D}} = 0, \qquad (1)$$

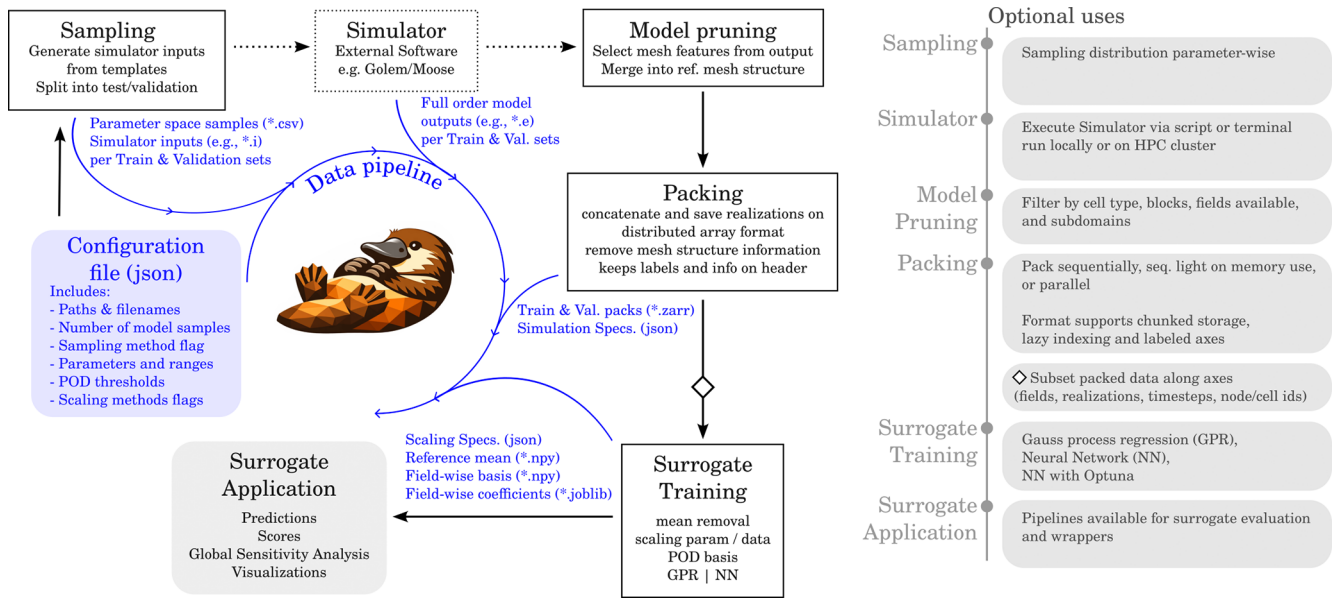


Figure 1. Model-to-surrogate workflow schematic. `Platypus` orchestrates each stage of the surrogate modeling pipeline, from parameter sampling and interfacing with an external simulator (in dotted box to emphasize its external character), through model curation, data management, and surrogate training, all guided by a centralized configuration file. The solid line boxes indicate modules for which platypus offers functionality. The data pipeline denoted by blue, indicates the output files obtained from a module at the time and available for the next steps. Available options at each stage are listed on the right side of the figure. Center: Platypus logo (OpenAI, 2025).

where p is the pore pressure, T is the temperature, t is the time, $\dot{\underline{\underline{\boldsymbol{\epsilon}}}}_{kk}$ is the total volumetric strain rate, $\dot{\underline{\underline{\boldsymbol{\epsilon}}}}^{\star}_{kk}$ is the inelastic strain rate, α is the Biot coefficient, M_{B} is the Biot modulus, β is the volumetric thermal expansion coefficient, and the subscripts f, s and b, correspond to the fluid, solid and bulk components, respectively. The Darcy velocity $\boldsymbol{q}_{\mathrm{D}}$ is given by

$$\boldsymbol{q}_{\mathrm{D}} = -\frac{1}{\mu}\underline{\underline{\boldsymbol{k}}}\left(\nabla p_{\mathrm{f}} - \rho_{\mathrm{f}}\boldsymbol{g}\right), \tag{2}$$

where $\underline{\underline{\boldsymbol{k}}}$ is the permeability tensor of the porous medium, μ is the fluid viscosity, ρ_{f} is the density of the fluid, and $\boldsymbol{g}$ the gravity vector.

The governing equation for heat transport results from energy balance of the porous-fracture system under thermal equilibrium conditions as

$$(\rho c)_{\mathrm{b}}\frac{\partial T}{\partial t} + \nabla\cdot\left(\rho_{\mathrm{f}}c_{\mathrm{f}}\boldsymbol{q}_{\mathrm{D}}T - \lambda_{\mathrm{b}}\nabla T\right) = 0, \tag{3}$$

where c is the specific heat capacity, $(\rho c)_{\mathrm{b}} = \phi\rho_{\mathrm{f}}c_{\mathrm{f}} + (1-\phi)\rho_{\mathrm{s}}c_{\mathrm{s}}$ is the bulk specific heat of the porous medium, ϕ is the porosity, $\lambda_{\mathrm{b}} = \phi\lambda_{\mathrm{f}} + (1-\phi)\lambda_{\mathrm{s}}$ is the bulk thermal conductivity.

The linear momentum balance equation of the bulk in terms of the effective Cauchy stress tensor $\underline{\underline{\boldsymbol{\sigma}}}'$ is given by

$$\nabla\cdot\left(\underline{\underline{\boldsymbol{\sigma}}}' - \alpha\, p_{\mathrm{f}}\underline{\underline{\mathbb{I}}}\right) + \rho_{\mathrm{b}}\boldsymbol{g} = 0. \tag{4}$$

There, $\underline{\underline{\mathbb{I}}}$ is the rank-two identity tensor, and ρ_{b} is the bulk density of the fluid–solid mixture ($\rho_{\mathrm{b}} = \phi\rho_{\mathrm{f}} + (1-\phi)\rho_{\mathrm{s}}$).

Further details about the governing equations, their derivation from mass, energy, and momentum conservation, and the underlying assumptions can be found in Cacace and Jacquey (2017). For a general review of coupled thermo-hydro-mechanical (and chemical) modeling of fractured porous media, see Vaezi et al. (2025), Zhang et al. (2023), Pandey et al. (2018) and references therein.

`Platypus` interfaces with a simulator by writing sampled parameter values into the solver's input, and reading the resulting ensemble of model solution snapshots. Supporting a different simulator amounts to adapting how sampled parameters are supplied to the solver, and how output fields are read back into the framework. The rest of the downstream modules remains unaffected. Minimal integration requires a function that exposes the model data to the framework. Deeper integration, such as automated execution of the solver or handling solver-specific mesh structures, requires more effort.

2.3 Model pruning

`Platypus` leverages PyVista (Sullivan and Kaszynski, 2019) and its VTK Exodus readers to read, filter, and visualize GOLEM model outputs for model curation and quantity of interest selection. This allows the user to work with entire model states or restrict the domain to the observation space alone.

The user can filter by predefined mesh block names (e.g., from volumetric lithologies or lower dimensional structures such as faults), cell types (e.g., triangles, tetrahedrons, hexahedrons), fields stored at nodal or cell level (e.g., temperature, pore pressure, stress components), merging the selection into a single unstructured mesh. The resulting reference mesh serves both as visualization output and as common spatial structure across the ensemble.

Model curation at this stage also includes identifying and discarding failed, incomplete or inconsistent simulations prior to ensemble assembly. Once the relevant fields are selected, mesh geometry and connectivity are dropped before packing, retaining only the field data and associated metadata and avoiding data duplication. Spatial information is not required for surrogate training or evaluation under the NIRB approach we employ, though a reference mesh is retained to visualize the spatial distribution of results.

2.4 Packing and metadata management

In this stage of data curation, we assemble and store the ensemble of simulations in a multidimensional, labelled data structure. Figure 2a illustrates the multidimensional character of the stored data, whose axes correspond to physical fields, time values, realization numbers, and spatial indices indicating whether field data was extracted node- or cell-wise. Beyond storage, this step enforces metadata consistency across simulations and provides structured, labelled access to the ensemble data.

`Platypus` leverages Zarr format with Xarray (Hoyer and Hamman, 2017) to store the n-dimensional array, and labels of their values across their axes to maintain interpretability. An additional advantage of using these libraries in the context of big scientific data lies on the distributed, chunked storage, lazy loading, and available compression algorithms for the packed data (e.g., Abernathey et al., 2021). The distributed storage is well-suited to parallel processing and cloud-native workflows. The lazy loading refers to having access to the ensemble dimensions without loading it all on memory, which is particularly helpful for extracting subdomains of the ensemble prior to loading in memory the data. To accommodate different computational environments, `Platypus` contains different packing strategies suited to both laptop and high-performance cluster settings.

2.5 Data preparation for surrogate training

`Platypus` applies a static correction and a normalization to improve conditioning and ensure that each feature has equal level of importance to model performance. The static correction, or "particular solution" (Swischuk et al., 2019), is substracted from the data to isolate the perturbations to the system. In `Platypus`, the user can choose this term as either the inital conditions of the simulation, or the node/cell-wise mean across realizations and time. `Platypus` supports standarization (Z-score normalization) and min-max-scaling. The computed scaling parameters are stored in JSON format to enable consistent forward and inverse scaling when evaluating the surrogate. The user can select a subset of the ensemble along any of its axes, on which both correction and scaling parameters are computed and the surrogate is trained.

2.6 Surrogates via non-intrusive reduced-basis method

In `Platypus` we adopt a hybrid surrogate approach, the non-intrusive reduced-basis method (Degen et al., 2022, 2023; Hesthaven and Ubbiali, 2018). Figure 2b shows the NIRB method, which consists of a decoupled offline and online phases. In the offline phase, the method uses snapshot collections from the full order models to extract a reduced basis through a proper orthogonal decomposition (POD) to capture characteristic physical behaviour, and uses machine learning to determine the weights of these basis functions. After the offline phase, the online phase is used to obtain efficient on-demand state predictions (Hesthaven and Ubbiali, 2018; Degen et al., 2022). This hybrid physics-based machine learning method enables the construction of robust, scalable, and interpretable surrogate models that approximate high-fidelity physical simulations without modifying the governing equations or the underlying solver.

The reduced solution u_{rb} for a set of parameters γ is approximated as the linear combination of basis functions,

$$u_{\mathrm{rb}}(\gamma) = \sum_{i=1}^{n} \psi_i \Theta_{\mathrm{rb}}^{(i)}(\gamma), \tag{5}$$

where ψ_i are the POD basis functions and $\Theta_{\mathrm{rb}}^{(i)}$ are the corresponding reduced coefficients evaluated for γ. The basis functions are truncated by imposing a threshold ϵ to a cumulative energy criterion defined by the singular values σ_i as

$$\frac{\sum_{i=1}^{n} \sigma_i^2}{\sum_{j=1}^{n_{\mathrm{tot}}} \sigma_j^2} \geq 1 - \epsilon, \tag{6}$$

where n_{tot} is the total number of training snapshots. This criterion selects the n most energetically significant basis functions. In `Platypus`, this criterion is handled at the configuration file, referred to as POD threshold.

`Platypus` includes two options to compute the coefficients. The first option is using Gaussian process regression, a non-parametric Bayesian approach that models input-output relations through a normal distribution prior, requiring few hyperparameters and yielding optimal coefficient estimates (Guo and Hesthaven, 2018; Lindner et al., 2025). It is well-suited to small scenarios and processes with relatively linear parameter variations (Raissi et al., 2017), with limited scalability (which has motivated efforts to overcome it; Liu et al., 2020), making it less appropriate for highly nonlinear processes. The second option involves estimating the

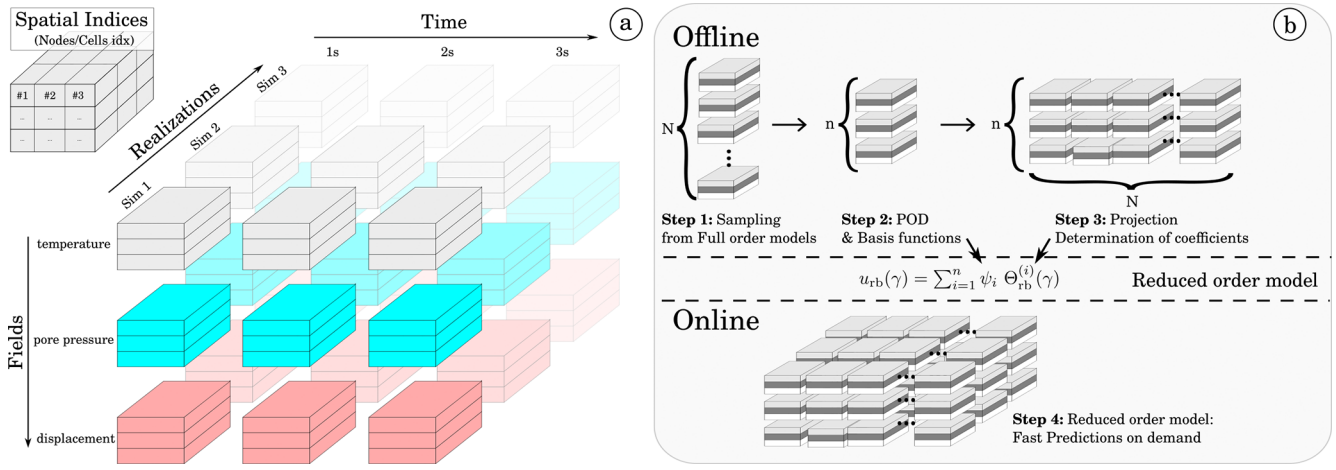


Figure 2. Data management and surrogate modeling via non-intrusive reduced-basis (NIRB) approach. **(a)** Multidimensional labelled data structure for ensemble storage, organized along axes representing physical fields, time steps, simulation realizations, and spatial indices (node- or cell-wise). **(b)** Surrogate construction via NIRB from snapshot collections from full order model. The method consists of two decoupled offline and online phases. In the offline phase, it uses an ensemble of snapshots to extract a reduced basis through proper orthogonal decomposition (POD), and uses machine learning to determine the weights of the basis functions. The online phase is then used to obtain fast predicitons on demand. Modified from Degen et al. (2025b).

coefficients via neural networks using, in our case, TensorFlow (Abadi et al., 2015; TensorFlow Developers, 2026). The latter is rather preferred for coeficient estimation from non-linear processes (Degen et al., 2022, 2023). This option also features the use of Optuna for hyperparameter tuning (Akiba et al., 2019). The surrogate training in `Platypus` adopts the NIRB implementation used and analysed in previous applications (Degen et al., 2025a, 2023, 2024, 2022; Santoso et al., 2022). The cost for this offline phase is dependent on the chosen QoI, degrees of freedom and complexity of the field to be reproduced, as well as timesteps and POD threshold, and surrogate training method.

2.7 Global sensitivity analysis pipelines

At the online stage, the surrogate model can be evaluated orders of magnitude faster than full-order simulations, enabling both real-time parameter exploration and large-scale statistical analysis. `Platypus` includes pipelines to compute variance-based Sobol sensitivity indices via SALib (Iwanaga et al., 2022; Herman and Usher, 2017). This allows the user to assess the influence of model parameters on selected quantities of interest, helping to reduce the parameter space for applications such as inversion, optimization, and uncertainty quantification. The Sobol method requires no a priori knowledge of parameter distributions and decomposes the model output variance into direct parameter effects (first-order indices), parameter interactions (higher-order indices), and total influence (total-order indices).

While a global sensitivity analysis (SA) is computationally demanding, it is a natural preparation step for model calibration (Dorward et al., 2024). Taking into consideration the information from parameter interactions significantly reduces the dimensionality of the problem, directly benefiting subsequent calibration and inversion schemes.

3 Examples of applications

We present two applications of `Platypus` demonstrating its use across time-dependent coupled hydrothermal and thermo-hydro-mechanical processes. The first example is a simple reservoir benchmark, where the trained surrogate is used to perform a global sensitivity analysis targeting material parameters. The second one is an application to the Groß Schönebeck geothermal site, featuring a multistage injection scenario, where we demonstrate how the workflow can be focused on a specific region of interest through model pruning.

3.1 Simple geothermal reservoir benchmark

The simple reservoir scenario consists of a synthetic geothermal doublet embedded in a low permeability formation at approximately 4 km depth with induced fractures at the two operational wells, designed to capture the thermohydraulic response to geothermal operations. We use this scenario to demonstrate a typical `Platypus` workflow, covering parameter sampling, model curation, ensemble assembly, surrogate construction, and global sensitivity analysis.

In this transient TH model, we vary porosity, permeability, density, heat capacity, and thermal conductivity of the solid media over ranges specified in Table 1, while all remaining material parameters are held constant at their reference values. The model domain consists of a 500 m cubic porous reservoir with a background temperature of 150 °C. Water is

Table 1. Parameter subspace for the simple geothermal reservoir ensemble.

Parameter	Symbol	Min–Max	Unit
Porosity	ϕ_{dom}	0.02–0.2	–
Permeability	k_{dom}	10^{-16}–10^{-15}	m^2
Density	ρ_{dom}	2400–2800	kg m^{-3}
Heat capacity	c_{dom}	900–1000	$\text{J kg}^{-1}\,\text{K}^{-1}$
Thermal cond.	λ_{dom}	2–4	$\text{W m}^{-1}\,\text{K}^{-1}$

Table 2. Parameter subspace for the Groß Schönebeck ensemble. Symbol subscripts indicate lithological unit: Elbe base sandstone I, and volcanic layer.

Parameter	Symbol	Min–Max	Unit
Porosity	ϕ_{Elbe}	0.135–0.165	–
Porosity	ϕ_{volc}	0.0045–0.0055	–
Permeability	k_{Elbe}	1.152 –1.408 ($\times 10^{-15}$)	m^2
Permeability	k_{volc}	0.8883–1.0857 ($\times 10^{-16}$)	m^2

injected at a temperature of 70 °C through an injection well at a rate of $3\,\text{L s}^{-1}$, with an equal production rate. The scenario has an operation time of 100 years. We sample the training ensemble of 200 realizations via Latin Hypercube sampling, and draw a validation ensemble of 40 realizations by random sampling, applying a logarithmic mapping to the permeability to avoid sampling bias. We run each simulation for 100 years with 100 saved timesteps.

We curate the ensemble by retaining pore pressure and temperature fields node-wise over the full volumetric domain, filtering for tetrahedral cell types and discarding fault surfaces and line elements. We then remove the training set mean from both training and validation sets, and apply standard scaling to the parameter sets and min-max scaling to the field data. POD basis extraction at a tolerance of 10^{-4} yields 14 basis functions for the pore pressure and 51 for the temperature. We estimate the reduced coefficients via neural networks with hyperparameter optimization through Optuna.

In the online stage, we compare the surrogate predictions against the validation set and use them to accelerate the evaluations for global SA. Figure 3a–b show a the temperature field after 100 years comparing the full order simulation output and the surrogate prediction. Figure 3c shows their difference, where blue marks regions where the surrogate underpredicts and red marks regions of over-prediction, with residuals below expected temperature measurement errors. Over all validation realizations, timesteps and spatial degrees of freedom, the relative L_2 norm is 9.45×10^{-4}. We then evaluate the surrogate model 1 572 864 times to compute Sobol sensitivity indices, following a Saltelli's sampling scheme (using 2^{17} samples, as described in Herman and Usher, 2017; Saltelli, 2002), on the domain averaged temperature, identifying ρ_{dom}, and λ_{dom} as the most influential parameters by their first-order indices.

3.2 Groß Schönebeck geothermal reservoir model

The Geothermal Research Platform Groß Schönebeck in Northern Germany is a laboratory for the development of geothermal technologies in hydrothermal reservoir rock with low permeability. Two research wells form a geothermal doublet providing access to water-bearing horizons at depths between 3.9 and 4.4 km at temperatures of 150 °C (Blöcher et al., 2016).

To study the multi-injection stimulation performed at this site, in which large volume of injected water at high flow states triggered micro-seismicity (Blöcher et al., 2018; Blöcher et al., 2016; Kwiatek et al., 2010; Moeck et al., 2009; chronology and references in Christi et al., 2025), Cacace et al. (2021) presented and validated a coupled thermo-hydro-mechanical model of the geothermal site, finding seismic rates sensitivity to fluid-pressure rates, temperature variations, and tectonic conditions.

In this example, we adopt the model from Cacace et al. (2021) to demonstrate how `Platypus` can be used to extract specific subdomains and lithological units together with fault features from the full-order model (location shown in Fig. 4a, b). We further use the extracted subdomain to train a surrogate of the pore pressure field, demonstrating that the workflow generalizes to focused domains and thermo-hydro-mechanical (THM) processes. In contrast to the simple reservoir benchmark, this scenario evolves over timescales of days, emphasizing different dominant physical processes.

The model geometry consists of a $1.6\,\text{km} \times 1.6\,\text{km} \times 0.51\,\text{km}$ reservoir. We vary the permeability and porosity for two lithological units listed in Table 2, maintaining the reference anisotropy ratios for the transversely isotropic permeability, with all remaining parameters held constant at their reference values. This choice of parameter space follows the most influential parameters identified in Degen et al. (2022). We sample a training ensemble of 200 realizations via LHS and a validation set of 50 realizations by random sampling. POD extraction at a tolerance of 5×10^{-4} results in 28 basis functions for pore pressure and 5 for temperature, and we estimate the reduced coefficients via neural networks with hyperparameter tuning. The relative L_2 norm across timesteps, realizations and spatially distributed pore pressure field yields a value of 1.34×10^{-2}. Figure 4c shows the pore pressure validation, demonstrating that `Platypus` can be used to construct an accurate surrogate on a spatial subdomain.

4 Discussion

Surrogates offer a computationally efficient alternative to full-order simulations, enabling large-scale statistical analysis. The examples in the previous section illustrate the flex-

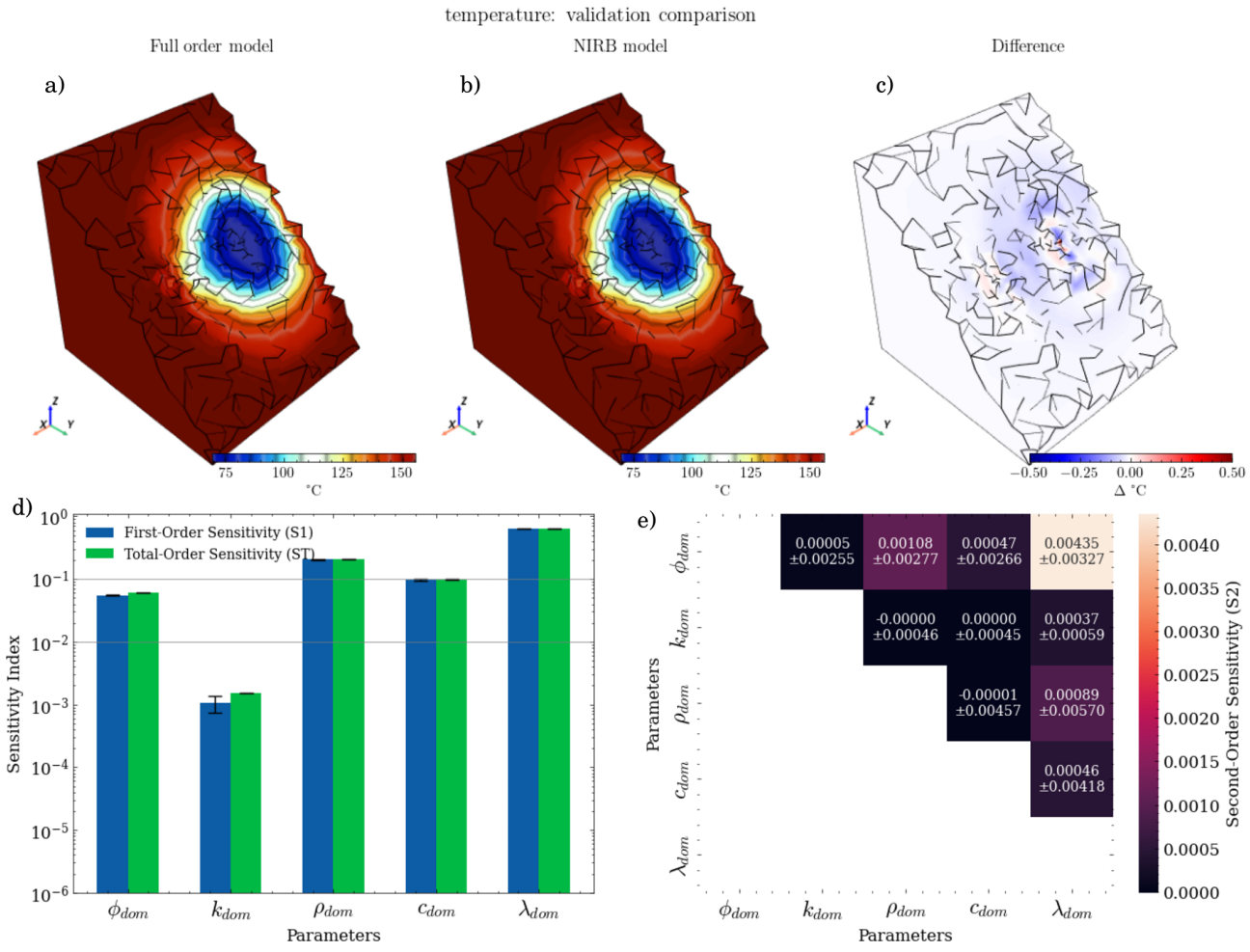


Figure 3. Surrogate validation and global sensitivity analysis for the simple geothermal reservoir benchmark. Temperature distribution of the **(a)** full finite element solution using GOLEM and **(b)** the NIRB surrogate prediction for a validation realization, depicting a geothermal doublet configuration. **(c)** Difference between the two solutions, with residuals of a fraction of °C across the domain, at a simulation time of 100 years. The surrogate model is then evaluated 1 572 864 times to perform a global sensitivity analysis on the domain averaged temperature. **(d)** First and Total-order Sobol sensitivity indices for the five material parameters of the reservoir domain. **(e)** Second-order sensitivity indices quantifying the combined influence of parameter pairs. Interaction effects remain small relative to first-order effects.

ibility of `Platypus` in handling the chain of challenges involved in curating an ensemble and constructing a reduced model for a given scenario.

More broadly, `Platypus` addresses a recurring need in reservoir engineering: ensuring consistency and traceability across ensemble-based modelling studies. The data pipeline shown in Fig. 1 collects the data and metadata exchanged between workflow steps, facilitating documentation and reproduction. This structure supports adoption across teams and projects, and provides a practical framework around which guidelines and recommended practices for ensemble-based reservoir modelling can be developed and shared.

The modularity of the package allows individual components to be adapted or replaced to suit specific modelling requirements, and can be integrated into external workflows independently. Within this framework, the focus shifts to identifying the quantity of interest that best sharpens the scientific or operational question around a given scenario.

The choice of quantity of interest can be computed at different stages of the workflow in Fig. 1. The user can compute it at run time directly from the solver, extract it from the packed ensemble, or derive it from the surrogate output. Computing within the solver itself is advantageous when it involves material parameters that vary in time, or fields too large to store in memory (e.g., the full stress tensor). When all required fields are available at the packing stage, the quantity of interest can be queried at any point in space and time. When deriving the quantity of interest from the surrogate, the user should account for potential propagation of numerical errors from each contributing field into the final quantity.

A practical consideration when assembling the ensemble is to pack a large model domain from the outset, even when

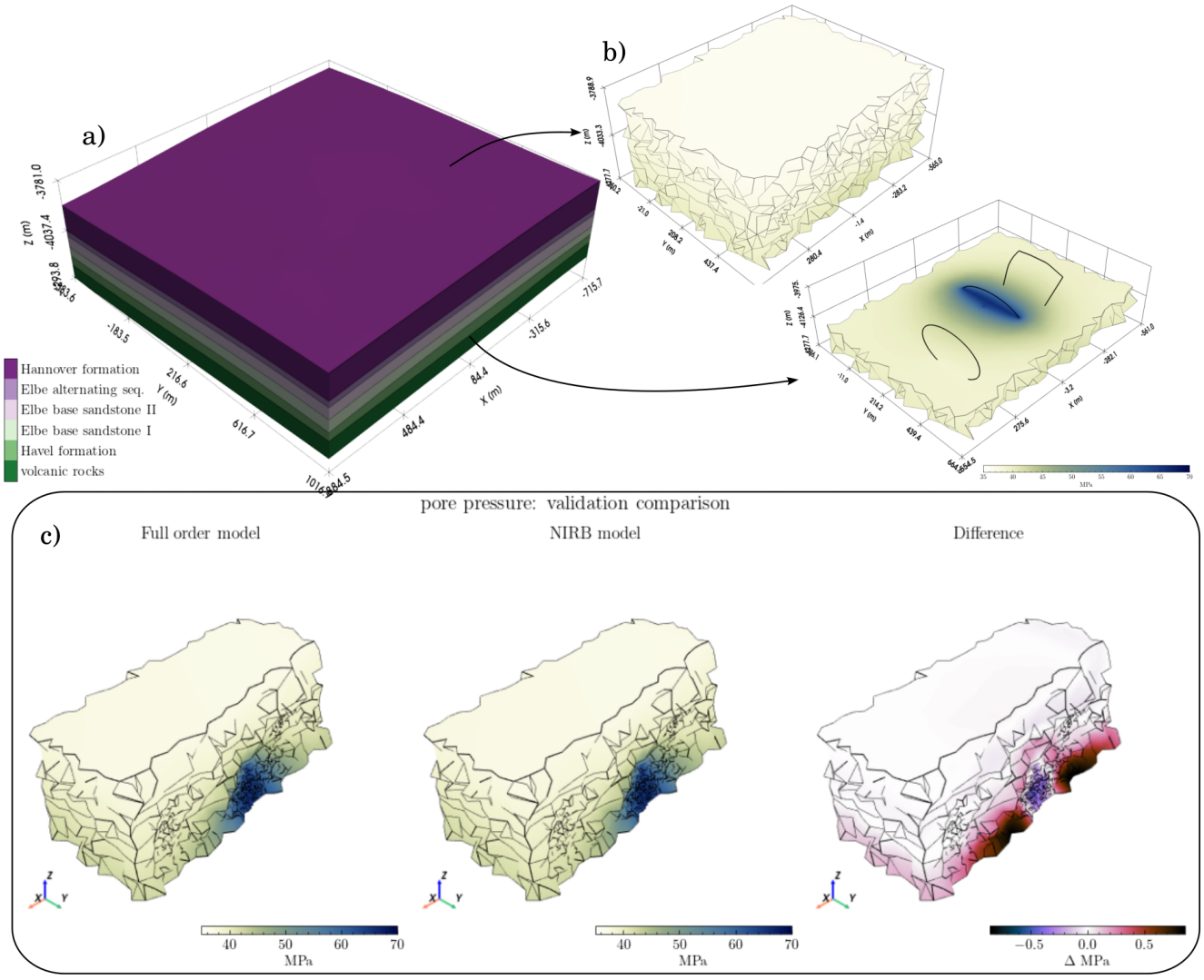


Figure 4. Groß Schönebeck geothermal site and surrogate validation. **(a)** Full 3D reservoir model (adopted from Cacace et al., 2021), composed of six sedimentary and volcanic units. **(b)** Model pruning application: the full domain is clipped to isolate the region of interest around the injection zone and simplified fault geometries. **(c)** Pore pressure validation comparing the full finite element solution, the NIRB surrogate prediction, and their difference at a simulation time of 1.75 d.

the intended analysis targets a specific subdomain. The format of the stored ensemble allows to query any subset of the ensemble (e.g., lithological units, clipped regions between isodepths, or lower dimensional features such as faults and wells), which reduces storage redundancy, preserving the flexibility to revisit different regions of interest without rerunning the data pipeline. The reference mesh retained after model pruning further supports this, as it provides the spatial index needed to map any queried subset back to its physical location in the model.

Limitations and Outlook

The workflow steps in `Platypus` described above represent a functional but extensible baseline. Several directions for future development are outlined here.

At the sampling stage, the current implementation supports Latin Hypercube and random sampling. Potential extensions include low-discrepancy sequences (Halton, 1960; Niederreiter, 1992; Sobol, 1998), as well as adaptive sampling schemes (e.g., Liu et al., 2018), which iteratively refine the parameter space coverage based on model responses.

Adapting `Platypus` to alternative solvers, such as finite volumes or finite differences codes, would require only a modified model pruning module to extract transient output fields, ensure consistent indexing across the realizations, and

assemble the ensemble of simulation data. For simulations that involve adaptive mesh refinement, where the number of nodes varies across time and realizations, projection methods that accommodate mesh variability exist (Gräßle et al., 2019; Alla et al., 2021). Alternatively, one may consider projecting onto a common reference mesh solution.

The surrogate training stage is currently based on the NIRB approach with POD for basis extraction, and Gaussian process regression or neural networks for coefficient estimation. This could be extended to alternative reduction methodologies, different machine learning backends, or problem-specific constraints on data handling.

Finally, while `Platypus` supports an arbitrary number of parameters and output fields, a large parameter space will demand a large ensemble to adequately capture the physical behaviour of the model, which in turn increases the computational cost of both simulation and surrogate training.

5 Conclusions

We have presented `Platypus`, an open-source, user-friendly Python package that integrates parameter sampling, model curation, and surrogate training for ensemble-based reservoir modelling. The examples demonstrate its applicability to coupled time-dependent hydrothermal and thermo-hydro-mechanical processes, where surrogate models reproduce full-order simulation outputs at a fraction of the computational cost, enabling global sensitivity analyses with over a million surrogate evaluations. Already adopted in geothermal modelling workflows, `Platypus` provides a structured and reproducible pipeline that ensures consistency and traceability across ensemble-based studies, supporting adoption across teams and projects. Its modular design allows individual components to be adapted, extended, or integrated into external workflows, making it applicable to a wider range of process-based simulators and subsurface applications.

Code and data availability. The `Platypus` Python package including model and input data for the synthetic simple geothermal reservoir is publicly available via a Zenodo repository (https://doi.org/10.5281/zenodo.20734438, Hayek et al., 2026). Field data and model associated to Groß Schönebeck are available upon reasonable request.

Author contributions. JNH: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing – original draft, writing – review and editing; MC: conceptualization, resources, methodology, formal analysis, writing – review and editing; MF: conceptualization, validation, writing – review and editing; YG: conceptualization, validation, writing – review and editing; DD: conceptualization, formal analysis, funding acquisition, project administration, resources, supervision, writing – original draft, writing – review and editing. All authors read and approved the final manuscript.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. The authors bear the ultimate responsibility for providing appropriate place names. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Special issue statement. This article is part of the special issue "European Geosciences Union General Assembly 2026, EGU Division Energy, Resources & Environment (ERE)". It is a result of the EGU General Assembly 2026, Vienna, Austria & Online, 3–8 May 2026.

Acknowledgements. This work utilized high-performance computing resources made possible by funding from the Ministry of Science, Research and Culture of the State of Brandenburg (MWFK) and are operated by the IT Services and Operations unit of the GFZ Helmholtz Centre for Geosciences.

The authors are grateful to Nimisha A. George, Cristian E. Siegel, and Türkan A. Bayter, for their early adoption of the package in their own workflows and for their valuable questions, which directly contributed to improvements in the software and its documentation.

Financial support. This research has been supported by the Bundesministerium für Forschung, Technologie und Raumfahrt through the project PBML-HM (grant no. 16lS24062).

The article processing charges for this open-access publication were covered by the GFZ Helmholtz Centre for Geosciences.

Review statement. This paper was edited by Thanushika Gunatilake and reviewed by Ana Gonzalez-Nicolas Alvarez and one anonymous referee.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Watten-

berg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, TensorFlow, https://www.tensorflow.org/ (last access: 19 June 2026), 2015.

Abernathey, R. P., Augspurger, T., Banihirwe, A., Blackmon-Luca, C. C., Crone, T. J., Gentemann, C. L., Hamman, J. J., Henderson, N., Lepore, C., McCaie, T. A., Robinson, N. H., and Signell, R. P.: Cloud-Native Repositories for Big Scientific Data, Comput. Sci. Eng., 23, 26–35, https://doi.org/10.1109/MCSE.2021.3059437, 2021.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, 2623–2631, ISBN 9781450362016, https://doi.org/10.1145/3292500.3330701, 2019.

Alla, A., Gräßle, C., and Hinze, M.: Time Adaptivity in Model Predictive Control, J. Sci. Comput., 90, 12, https://doi.org/10.1007/s10915-021-01647-0, 2021.

Benner, P., Gugercin, S., and Willcox, K.: A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems, SIAM Rev., 57, 483–531, https://doi.org/10.1137/130932715, 2015.

Blöcher, G., Reinsch, T., Henninges, J., Milsch, H., Regenspurg, S., Kummerow, J., Francke, H., Kranz, S., Saadat, A., Zimmermann, G., and Huenges, E.: Hydraulic history and current state of the deep geothermal reservoir Groß Schönebeck, Geothermics, 63, 27–43, https://doi.org/10.1016/j.geothermics.2015.07.008, 2016.

Blöcher, G., Cacace, M., Jacquey, A. B., Zang, A., Heidbach, O., Hofmann, H., Kluge, C., and Zimmermann, G.: Evaluating Micro-Seismic Events Triggered by Reservoir Operations at the Geothermal Site of Groß Schönebeck (Germany), Rock Mech. Rock Eng., 51, 3265–3279, 2018.

Cacace, M. and Jacquey, A. B.: Flexible parallel implicit modelling of coupled thermal–hydraulic–mechanical processes in fractured rocks, Solid Earth, 8, 921–941, https://doi.org/10.5194/se-8-921-2017, 2017.

Cacace, M., Hofmann, H., and Shapiro, S. A.: Projecting seismicity induced by complex alterations of underground stresses with applications to geothermal systems, Sci. Rep., 11, 23560, https://doi.org/10.1038/s41598-021-02857-0, 2021.

Christi, L. F., Sass, I., Norden, B., Blöcher, G., Zimmermann, G., and Hofmann, H.: Repurposing of hydrocarbon wells for Enhanced Geothermal System (EGS) development, Geothermics, 128, 103268, https://doi.org/10.1016/j.geothermics.2025.103268, 2025.

Degen, D. and Wellmann, F.: Guidelines for Conducting Sensitivity Analyses in Process Simulations for Solid Earth Geosciences, Math. Geosci., https://doi.org/10.1007/s11004-025-10267-5, 2026.

Degen, D., Cacace, M., and Wellmann, F.: 3D multi-physics uncertainty quantification using physics-based machine learning, Sci. Rep., 12, 17491, https://doi.org/10.1038/s41598-022-21739-7, 2022.

Degen, D., Caviedes Voullième, D., Buiter, S., Hendricks Franssen, H.-J., Vereecken, H., González-Nicolás, A., and Wellmann, F.: Perspectives of physics-based machine learning strategies for geoscientific applications governed by partial differential equations, Geosci. Model Dev., 16, 7375–7409, https://doi.org/10.5194/gmd-16-7375-2023, 2023.

Degen, D., Cacace, M., and Wellmann, F.: Exploring physics-based machine learning for geothermal applications, in: 49th Workshop on Geothermal Reservoir Engineering, 1–9, https://pangea.stanford.edu/ERE/db/IGAstandard/record_detail.php?id=36316 (last access: 19 June 2026), 2024.

Degen, D., Kumar, A., Scheck-Wenderoth, M., and Cacace, M.: Exploiting physics-based machine learning to quantify geodynamic effects – insights from the Alpine region, Geosci. Model Dev., 18, 9219–9236, https://doi.org/10.5194/gmd-18-9219-2025, 2025a.

Degen, D., Ziegler, M., Heidbach, O., Henk, A., Reiter, K., and Wellmann, F.: About the trustworthiness of physics-based machine learning – considerations for geomechanical applications, Solid Earth, 16, 477–502, https://doi.org/10.5194/se-16-477-2025, 2025b.

Dorward, H., Knowles, D. M., Demir, E., Mostafavi, M., and Peel, M. J.: Calibration and surrogate model-based sensitivity analysis of crystal plasticity finite element models, Mater. Design, 247, 113409, https://doi.org/10.1016/j.matdes.2024.113409, 2024.

Gräßle, C., Hinze, M., Lang, J., and Ullmann, S.: POD model order reduction with space-adapted snapshots for incompressible flows, Adv. Comput. Math., 45, 2401–2428, https://doi.org/10.1007/s10444-019-09716-7, 2019.

Guo, M. and Hesthaven, J. S.: Reduced order modeling for nonlinear structural analysis using Gaussian process regression, Comput. Method. Appl. M., 341, 807–826, https://doi.org/10.1016/j.cma.2018.07.017, 2018.

Halton, J. H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numer. Math., 2, 84–90, https://doi.org/10.1007/BF01386213, 1960.

Harbour, L., Giudicelli, G., Lindsay, A. D., German, P., Hansel, J., Icenhour, C., Li, M., Miller, J. M., Stogner, R. H., Behne, P., Yankura, D., Prince, Z. M., DeChant, C., Schwen, D., Spencer, B. W., Tano, M., Choi, N., Wang, Y., Nezdyur, M., Miao, Y., Hu, T., Kumar, S., Matthews, C., Langley, B., Nobre, N., Blair, A., MacMackin, C., Rocha, H. B., Palmer, E., Carter, J., Meier, J., Slaughter, A. E., Andrš, D., Carlsen, R. W., Kong, F., Gaston, D. R., and Permann, C. J.: 4.0 MOOSE: Enabling massively parallel Multiphysics simulation, SoftwareX, 31, 102264, https://doi.org/10.1016/j.softx.2025.102264, 2025.

Hayek, J. N., Cacace, M., Faber, M., Gruzdeva, Y., and Degen, D.: Platypus: A Python package for automated surrogate-assisted analysis of process-based simulations via the non-intrusive reduced-basis method, Zenodo [code], https://doi.org/10.5281/zenodo.20734438, 2026.

Herman, J. and Usher, W.: SALib: An open-source Python library for Sensitivity Analysis, Journal of Open Source Software, 2, https://doi.org/10.21105/joss.00097, 2017.

Hesthaven, J. and Ubbiali, S.: Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys., 363, 55–78, https://doi.org/10.1016/j.jcp.2018.02.037, 2018.

Hesthaven, J. S., Rozza, G., and Stamm, B.: Reduced Basis Methods, Springer International Publishing, Cham, 27–43, ISBN 978-3-319-22470-1, https://doi.org/10.1007/978-3-319-22470-1_3, 2016.

Hoyer, S. and Hamman, J.: xarray: N-D labeled Arrays and Datasets in Python, Journal of Open Research Software, https://doi.org/10.5334/jors.148, 2017.

Iwanaga, T., Usher, W., and Herman, J.: Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses, Socio-Environmental Systems Modelling, 4, 18155, https://doi.org/10.18174/sesmo.18155, 2022.

Jacquey, A. B. and Cacace, M.: GOLEM, a MOOSE-based application, Zenodo [code], https://doi.org/10.5281/zenodo.999401, 2017.

Kwiatek, G., Bohnhoff, M., Dresen, G., Schulze, A., Schulte, T., Zimmermann, G., and Huenges, E.: Microseismicity induced during fluid-injection: A case study from the geothermal site at Groß Schönebeck, North German Basin, Acta Geophys., 58, 995–1020, https://doi.org/10.2478/s11600-010-0032-7, 2010.

Lindner, N., Degen, D., Grayver, A., and Wellmann, F.: Sensitivity analysis using physics-based machine learning: an example from surrogate modelling for magnetotellurics, Geophys. J. Int., 242, ggaf166, https://doi.org/10.1093/gji/ggaf166, 2025.

Liu, H., Ong, Y.-S., and Cai, J.: A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design, Struct. Multidiscip. O., 57, 393–416, https://doi.org/10.1007/s00158-017-1739-8, 2018.

Liu, H., Ong, Y.-S., Shen, X., and Cai, J.: When Gaussian Process Meets Big Data: A Review of Scalable GPs, IEEE T. Neur. Net. Lear., 31, 4405–4423, https://doi.org/10.1109/TNNLS.2019.2957109, 2020.

Loh, W.-L.: On Latin hypercube sampling, Ann. Stat., 24, 2058–2080, https://doi.org/10.1214/aos/1069362310, 1996.

McKay, M. D., Beckman, R. J., and Conover, W. J.: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, Technometrics, 21, 239–245, https://doi.org/10.2307/1268522, 1979.

Moeck, I., Kwiatek, G., and Zimmermann, G.: Slip tendency analysis, fault reactivation potential and induced seismicity in a deep geothermal reservoir, J. Struct. Geol., 31, 1174–1182, https://doi.org/10.1016/j.jsg.2009.06.012, 2009.

Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, Society for Industrial and Applied Mathematics, https://doi.org/10.1137/1.9781611970081, 1992.

OpenAI: Minimalistic platypus logo transitioning into a low poly geometric style, AI-generated image, ChatGPT (DALL**.**E), https://chatgpt.com (last access: 16 February 2026), 2025.

Pandey, S., Vishal, V., and Chaudhuri, A.: Geothermal reservoir modeling in a coupled thermo-hydro-mechanical-chemical approach: A review, Earth-Sci. Rev., 185, 1157–1169, https://doi.org/10.1016/j.earscirev.2018.09.004, 2018.

Raissi, M., Perdikaris, P., and Karniadakis, G. E.: Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations, arXiv [preprint], https://doi.org/10.48550/arXiv.1711.10561, 2017.

Saltelli, A.: Making best use of model evaluations to compute sensitivity indices, Comput. Phys. Commun., 145, 280–297, https://doi.org/10.1016/S0010-4655(02)00280-1, 2002.

Santoso, R., Degen, D., Cacace, M., and Wellmann, F.: State-of-the-art physics-based machine learning for hydro-mechanical simulation in geothermal applications, European Geothermal Congress 2022, https://doi.org/10.24406/publica-1122, 2022.

Sobol, I.: On quasi-Monte Carlo integrations, Math. Comput. Simulat., 47, 103–112, https://doi.org/10.1016/S0378-4754(98)00096-2, 1998.

Sullivan, B. and Kaszynski, A.: PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK), Journal of Open Source Software, 4, 1450, https://doi.org/10.21105/joss.01450, 2019.

Swischuk, R., Mainini, L., Peherstorfer, B., and Willcox, K.: Projection-based model reduction: Formulations for physics-based machine learning, Comput. Fluids, 179, 704–717, https://doi.org/10.1016/j.compfluid.2018.07.021, 2019.

TensorFlow Developers: TensorFlow (Versions v2.22.0-rc0), Zenodo [code], https://doi.org/10.5281/zenodo.4724125, 2026.

Vaezi, I., Yoshioka, K., De Simone, S., Gómez-Castro, B. M., Paluszny, A., Jalali, M., Berre, I., Rutqvist, J., Min, K.-B., Lei, Q., Makhnenko, R. Y., Hu, M., Tsang, C.-F., and Vilarrasa, V.: A review of thermo-hydro-mechanical modeling of coupled processes in fractured rock: From continuum to discontinuum perspective,, 17, 7460–7488, Journal of Rock Mechanics and Geotechnical Engineeringhttps://doi.org/10.1016/j.jrmge.2025.04.033, 2025.

Zhang, N., Luo, Z., Chen, Z., Liu, F., Liu, P., Chen, W., Wu, L., and Zhao, L.: Thermal–hydraulic–mechanical–chemical coupled processes and their numerical simulation: a comprehensive review, Acta Geotech., 18, 6253–6274, https://doi.org/10.1007/s11440-023-01976-4, 2023.