Engineering with Computers (2023) 39:4061–4076
https://doi.org/10.1007/s00366-023-01868-x

ORIGINAL ARTICLE

Reduced order model approaches for predicting the magnetic polarizability tensor for multiple parameters of interest

James Elgy[1] · **Paul D. Ledger**[1]

Received: 20 September 2022 / Accepted: 25 June 2023 / Published online: 15 July 2023

Abstract

The magnetic polarizability tensor (MPT) is an economical characterisation of a conducting magnetic object, which can assist with identifying hidden targets in metal detection. The MPT's coefficients depend on multiple parameters of interest including the object shape, size, electrical conductivity, magnetic permeability, and the frequency of excitation. The computation of the coefficients follow from post-processing an eddy current transmission problem solved numerically using high-order finite elements. To reduce the computational cost of constructing these characterisations for multiple different parameters, we compare three methods by which the MPT can be efficiently calculated for two-dimensional parameter sets, with different levels of code invasiveness. We compare, with numerical examples, a neural network regression of MPT eigenvalues with a projection-based reduced order model (ROM) and a neural network enhanced ROM (POD–NN) for predicting MPT coefficients.

Keywords Finite element method · Magnetic polarizability tensor · Metal detection · Object characterisation · Reduced order model · Neural networks

Article Highlights:

1. Rapid computations of object characterisations with varying material parameters to assist with metal detection.
2. A comparison between novel neural network and projection enhanced reduced order models for efficient computation.
3. Practical demonstrations of alternative methodologies including comparisons of computational cost.

1 Introduction

In recent years, there has been considerable interest in the characterisation of hidden conducting permeable objects by the magnetic polarizability tensor (MPT) and its applications to metal detection. The complex symmetric rank 2 MPT has been shown to offer an economical method of characterising conducting permeable objects [1–3], explicit formulae for calculating its 6 independent complex coefficients based on the object size, shape, electrical conductivity, magnetic permeability, and frequency of excitation have been derived [1, 2, 4], computational procedures proposed for its calculation [5] and apparatus for its measurement are advanced [6–9].

Key metal detection applications include in the discrimination between threat and non-threat objects in security screening using walk through metal detectors [3], whereby the early detection of threat objects (such as knives and firearms or components thereof) has the potential to reduce the likelihood of attacks and improve public safety. Further security applications include distinguishing between metallic clutter (e.g., ring-pulls, coins, shrapnel) and metallic components of hidden anti-personnel mines and unexploded ordnance [10]. Commercial applications, such as ensuring food safety screening, improving identification of metallic

These authors contributed equally to this work.

✉ James Elgy
J.Elgy@Keele.ac.uk

Paul D. Ledger
P.D.Ledger@Keele.ac.uk

[1] School of Computer Science and Mathematics, Keele University, Keele, Staffordshire ST5 5BG, UK

objects of significance in archaeological searches, and discriminating between real and counterfeit coinage at automated checkouts and vending machines are also of interest.

An approach for computing the MPT object characterisation as a function of exciting frequency, known as the MPT spectral signature, based on computing full-order model solutions at a small number of snapshot frequencies and using a (projected) proper orthogonal decomposition (POD) based reduced order model (ROM) [11] to predict the solution for other frequencies, has been proposed in [5] and implemented in the `MPT-Calculator` software. To compute the full order model solutions, the `NGSolve` high order finite element library [12–15] was used and a $\boldsymbol{H}$(curl) conforming discretisation on unstructured tetrahedral meshes was employed. The resulting characterisations have been shown to be in excellent agreement with practical measurements [9] for a wide range of object shapes. The `MPT-Calculator` tool has subsequently been used in combination with exact MPT scalings to generate dictionaries of realistic object characterisations [16], which, in turn, have been used for training machine learning classifiers for identifying possible threat and non–threat objects [17].

To be able to build larger dictionaries of MPT spectral signature object characterisations with increased variability in the object's material parameters, we introduce and compare three alternative novel ROMs in this paper. First, we extend the ROM presented in [5] to two parameters, namely frequency and permeability. Secondly, building on the approaches proposed in [18, 19], a regression-based POD is employed, which involves a neural network-based regression of information from the truncated singular value decomposition of the snapshot solution matrix to make predictions for new problem parameters. A recent extensive discussion of neural networks with applications in POD reduced order modelling is given in [20], covering artificial neural networks, physics informed neural networks, and feed-forward neural networks and the differences between them in the context of POD which provides further context to our methodology. Third, a neural network regression of the MPT eigenvalues is developed to predict MPT eigenvalues for new problem parameters. We then compare the accuracy and the computational performance of the three approaches. A further important aspect we consider is code invasiveness. POD approaches require access to the underlying finite element implementation, which may not be possible in many cases, for example using commercial closed–source software. On the other hand, POD-NN requires only access to computed solution vectors and a direct regression of the parameters requires no access to the underlying code.

This paper is organised as follows: Sect. 2 briefly reviews the mathematical formulation of the rank 2 MPT object characterisation of an isolated highly conducting magnetic object in a non–conducting background for the eddy current time–harmonic approximation to the Maxwell system. Section 3 recalls the *hp* finite element approximations to a transmission problem, which is used for calculating full-order model solutions, and Sect. 4 describes our proposed ROMs to accelerate this computation when evaluating for different material properties. This is followed, in Sect. 5, by numerical examples of the ROM approaches for an object with a known analytical solution for its MPT coefficients, and an object where there is no known analytical solution. Finally, concluding remarks and intended future work are provided in Sect. 6.

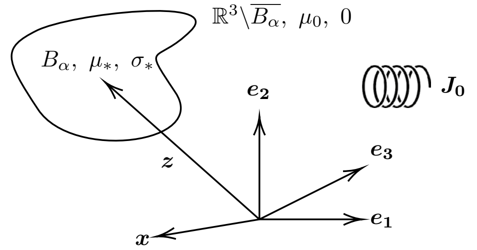


Fig. 1 Illustration of the conducting object B_α with position $\boldsymbol{z}$ and an arbitrary position vector $\boldsymbol{x}$ both with respect to the orthonormal coordinate system with basis vectors $\boldsymbol{e}_1, \boldsymbol{e}_2$, and $\boldsymbol{e}_3$

2 The Eddy Current Model and The Rank 2 MPT

We briefly recall the problem description from [2, 4, 5]. As illustrated in Fig. 1, our interest lies in characterising a highly conducting magnetic object, B_α set in an unbounded region of free space $B_\alpha^c := \overline{B_\alpha}\backslash\mathbb{R}^3$, where the overbar denotes the closure. Later, we will also use the overbar to denote the complex conjugate, however, it should be clear from the context as to which definition applies. We write $B_\alpha = \alpha B + \boldsymbol{z}$ so that the object can be described by a unit-sized object B placed at the origin, which is scaled by a size parameter $\alpha \ll 1$ (measured in m) and translated by $\boldsymbol{z}$. At a position $\boldsymbol{x}$, the material properties are

$$\mu_\alpha(\boldsymbol{x}) := \begin{cases} \mu_* & \boldsymbol{x} \in B_\alpha \\ \mu_0 & \boldsymbol{x} \in B_\alpha^c \end{cases}, \quad \sigma_\alpha(\boldsymbol{x}) := \begin{cases} \sigma_* & \boldsymbol{x} \in B_\alpha \\ 0 & \boldsymbol{x} \in B_\alpha^c \end{cases}, \tag{1}$$

where μ represents the permeability [Hm^{-1}] and σ represents the conductivity [Sm^{-1}], the subscript $*$ denotes their values inside B_α and the subscript 0 their values outside. The free space permeability is $\mu_0 := 4\pi \times 10^{-7}$ Hm^{-1} and we introduce the relative permeability $\mu_r := \mu_*/\mu_0$ inside the object.

An asymptotic formula has been established for the perturbation in magnetic field $(\boldsymbol{H}_\alpha - \boldsymbol{H}_0)(\boldsymbol{x})$ at positions $\boldsymbol{x}$ away from B_α as $\alpha \to 0$ when the object is placed in a time-varying low frequency magnetic background field $\boldsymbol{H}_0$ generated by an electric current source placed external to the body [2, 21]. This assumes that the eddy current approximation of the time harmonic Maxwell system has been applied, in which displacement currents are neglected, and is appropriate given the highly conducting nature of B_α and the low angular frequencies ω of the excitation. The form of this expansion is

$$\begin{aligned}\big(\boldsymbol{H}_\alpha - \boldsymbol{H}_0\big)(\boldsymbol{x})_i = &\big(\boldsymbol{D}_{\boldsymbol{x}}^2 G(\boldsymbol{x}, \boldsymbol{z})\big)_{ij} \\ &(\mathcal{M})_{jk}\big(\boldsymbol{H}_0(\boldsymbol{z})\big)_k + (\boldsymbol{R}(\boldsymbol{x}))_i,\end{aligned} \tag{2}$$

with $\boldsymbol{R}(\boldsymbol{x})$ denoting the residual, which satisfies $|\boldsymbol{R}(\boldsymbol{x})| \le C\alpha^4$ with C being a constant independent of α. In the above, $G(\boldsymbol{x}, \boldsymbol{z}) := 1/(4\pi|\boldsymbol{x} - \boldsymbol{z}|)$ is the free space Laplace Green's function and $\boldsymbol{D}_{\boldsymbol{x}}^2 G$ denotes the Hessian of G. The subscripts i, j, and k denote the component indices and Einstein summation convention is assumed. The complex symmetric rank 2 tensor $\mathcal{M} = (\mathcal{M})_{jk}\boldsymbol{e}_j \otimes \boldsymbol{e}_k$ is the MPT and can be decomposed as [1]

$$\mathcal{M} = \mathcal{N}^0 + \mathcal{R} + \mathrm{i}\mathcal{I} = \tilde{\mathcal{R}} + \mathrm{i}\mathcal{I}, \tag{3}$$

where $\mathrm{i} := \sqrt{-1}$, $\mathcal{N}^0(\alpha B, \mu_r)$ denotes its magnetostatic contribution, $\tilde{\mathcal{R}}(\alpha B, \omega, \sigma_*, \mu_r) = \mathcal{N}^0(\alpha B, \mu_r) + \mathcal{R}(\alpha B, \omega, \sigma_*, \mu_r)$ its frequency dependent real part and $\mathcal{I}(\alpha B, \omega, \sigma_*, \mu_r)$ its frequency dependent imaginary part. Their coefficients can be found from

$$\begin{aligned}(\mathcal{N}^0)_{ij} = &\alpha^3\delta_{ij}\int_B \big(1 - \mu_r{}^{-1}\big)\mathrm{d}\boldsymbol{\xi} \\ &+ \frac{\alpha^3}{4}\int_{B\cup B^c} \tilde{\mu}_r^{-1}\nabla\times\tilde{\boldsymbol{\theta}}_i^{(0)}\cdot\nabla\times\tilde{\boldsymbol{\theta}}_j^{(0)}\mathrm{d}\boldsymbol{\xi},\end{aligned} \tag{4a}$$

$$(\mathcal{R})_{ij} = -\frac{\alpha^3}{4}\int_{B\cup B^c}\tilde{\mu}_r^{-1}\nabla\times\boldsymbol{\theta}_i^{(1)}\cdot\nabla\times\overline{\boldsymbol{\theta}_j^{(1)}}\mathrm{d}\boldsymbol{\xi}, \tag{4b}$$

$$\begin{aligned}(\mathcal{I})_{ij} = &\frac{\alpha^3}{4}\int_B \nu\Big(\boldsymbol{\theta}_i^{(1)} + \big(\tilde{\boldsymbol{\theta}}_i^{(0)} + \boldsymbol{e}_i\times\boldsymbol{\xi}\big)\Big) \\ &\cdot\overline{\Big(\boldsymbol{\theta}_j^{(1)} + \big(\tilde{\boldsymbol{\theta}}_j^{(0)} + \boldsymbol{e}_j\times\boldsymbol{\xi}\big)\Big)}\mathrm{d}\boldsymbol{\xi}.\end{aligned} \tag{4c}$$

In (4), δ_{ij} is the Kronecker delta, $\nu := \alpha^2\sigma_*\mu_0\omega$, and $\tilde{\mu}_r(\boldsymbol{\xi}) = \mu_r$ inside B and $\tilde{\mu}_r(\boldsymbol{\xi}) = 1$ outside where $\boldsymbol{\xi}$ is chosen to be measured from an origin inside B. The vector field $\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^{(0)} + \boldsymbol{\theta}_i^{(1)} = \tilde{\boldsymbol{\theta}}_i^{(0)} + \boldsymbol{e}_i\times\boldsymbol{\xi} + \boldsymbol{\theta}_i^{(1)}$ is the solution to the transmission problem [1]:

$$\nabla\times\tilde{\mu}_r{}^{-1}\nabla\times\boldsymbol{\theta}_i - \mathrm{i}\nu\boldsymbol{\theta}_i = \mathrm{i}\nu\boldsymbol{e}_i\times\boldsymbol{\xi} \quad \text{in } B, \tag{5a}$$

$$\nabla\cdot\boldsymbol{\theta}_i = 0 \quad \text{in } B^c = \mathbb{R}^3\setminus\overline{B}, \tag{5b}$$

$$\nabla\times\nabla\times\boldsymbol{\theta}_i = \boldsymbol{0} \quad \text{in } B^c, \tag{5c}$$

$$\big[\boldsymbol{\theta}_i\times\boldsymbol{n}\big]_\Gamma = \boldsymbol{0} \quad \text{on } \Gamma := \partial B, \tag{5d}$$

$$\big[\tilde{\mu}_r^{-1}\nabla\times\boldsymbol{\theta}_i\times\boldsymbol{n}\big]_\Gamma = -2\big[\tilde{\mu}_r^{-1}\big]_\Gamma\boldsymbol{e}_i\times\boldsymbol{n} \quad \text{on } \Gamma, \tag{5e}$$

$$\boldsymbol{\theta}_i(\boldsymbol{\xi}) = \mathcal{O}\big(|\boldsymbol{\xi}|^{-1}\big) \quad \text{as } |\boldsymbol{\xi}|\to\infty, \tag{5f}$$

where $[\cdot]_\Gamma$ denotes the jump over Γ and $\boldsymbol{n}$ is the unit outward normal. The above problem can be split to form separate problems for $\tilde{\boldsymbol{\theta}}_i^{(0)}$ and $\boldsymbol{\theta}_i^{(1)}$, where the former is independent of ω and is a real vector field and the latter is a complex frequency-dependent vector field. Scaling results are available that allow the immediate calculation of $(\mathcal{M})_{ij}$ for new values of σ_* and α. The goal of this paper is to compare ROM approaches for rapidly predicting $(\mathcal{M})_{ij}$ for different μ_r and ω [5]. This in turn will aid with creating large dictionaries of object characterisations for training machine learning classifiers, which cannot be achieved through the application of scaling results.

3 Finite Element Approximation

As described in [5], by truncating the unbounded domain sufficiently far from B to create a finite computational domain Ω, replacing the far field condition (5f) with $\boldsymbol{n}\times\boldsymbol{\theta}_i = \boldsymbol{0}$ on $\partial\Omega$, circumventing the Coulomb type gauge $\nabla\cdot\boldsymbol{\theta}_i = 0$ in $\Omega\backslash\overline{B}$ with numerical regularisation (by solving a perturbed problem involving a small regularisation parameter ε), and employing a higher order $\boldsymbol{H}(\text{curl})$ conforming finite element approximation a discrete finite element approximation to the continuous weak forms for the $\tilde{\boldsymbol{\theta}}_i^{(0)}$ and $\boldsymbol{\theta}_i^{(1)}$ problems can be established. For both the $\tilde{\boldsymbol{\theta}}_i^{(0)}$ and $\boldsymbol{\theta}_i^{(1)}$ problems, an unstructured mesh of tetrahedral elements of size h is used to partition Ω and order p elements applied leading to a linear system of equations of the form

$$\mathbf{A}\mathbf{q}(\boldsymbol{\omega}) = \mathbf{r}, \tag{6}$$

where for $\boldsymbol{\theta}_i^{(1)}$, $\mathbf{A}\in\mathbb{C}^{N_{dof}\times N_{dof}}$ is a large sparse complex symmetric matrix, $\mathbf{q}(\boldsymbol{\omega})\in\mathbb{C}^{N_{dof}}$ is a parameter dependent solution with $\boldsymbol{\omega}$ indicating the list of model parameters to be varied, and $\mathbf{r}\in\mathbb{C}^{N_{dof}}$ is a known forcing vector. The situation is similar for the simpler $\boldsymbol{\theta}_i^{(0)}$ that involves real matrices. Once the solution to (6) has been established, the discrete approximation to $\boldsymbol{\theta}_i^{(1)}$ is recovered using

$$\boldsymbol{\theta}_i^{(1,hp)}(\boldsymbol{\omega}, \boldsymbol{\xi}) = \sum_{u=1}^{N_{dof}} \mathrm{q}_u^{(1)}(\boldsymbol{\omega})\mathbf{N}_u(\boldsymbol{\xi}), \tag{7}$$

where $\mathbf{N}_u(\boldsymbol{\xi})$ is a typical $\boldsymbol{H}(\mathrm{curl})$ conforming basis function and N_{dof} is the number of degrees of freedom in the finite element approximation. A similar approximation also applies for $\tilde{\boldsymbol{\theta}}_i^{(0,hp)}$. The approximate computation of $(\mathcal{M})_{ij}$ then follows by replacing $\boldsymbol{\theta}_i^{(1)}$ with $\boldsymbol{\theta}_i^{(1,hp)}$, $\tilde{\boldsymbol{\theta}}_i^{(0)}$ with $\tilde{\boldsymbol{\theta}}_i^{(0,hp)}$, and B^c with $\Omega\backslash\overline{B}$ in (4).

4 Reduced Order Approaches

The repeated solution of equation (6) for different model parameters $\boldsymbol{\omega}$ is computationally expensive. In [5], a projection based POD approach was developed (called PODP) in which the solution of (6) for new $\mathbf{q}(\boldsymbol{\omega})$ is replaced by solving a smaller projected linear system of equations of size $M \times M$ where $M \ll N_{dof}$. The reduced system was obtained by extracting the modal behaviour from a small number of solution snapshots and using Galerkin projection. The implementation in [5] was limited to the case where $\boldsymbol{\omega} = [\omega]$ and we consider the extension to $\boldsymbol{\omega} = [\omega, \mu_r]$.

As alternatives to this approach, we also consider two other approaches that are less intrusive to the software as they do not need direct access to (decompositions of) both $\mathbf{A}$ and $\mathbf{r}$. In the first of these alternatives, we employ a technique equivalent to POD-NN used by Hesthaven and Ubbiali [19], which is built by performing a neural network regression, leading to an approximation to $\mathbf{q}(\boldsymbol{\omega})$ for new $\boldsymbol{\omega}$. The approximate $\tilde{\boldsymbol{\theta}}_i^{(0)}$ and $\boldsymbol{\theta}_i^{(1)}$ then are obtained from (7) and the MPT coefficients are obtained as before by a simple post-processing. In the second of the alternatives, we consider a direct neural network regression of the MPT coefficients to predict the MPT coefficients for new $\boldsymbol{\omega}$. PODP and POD–NN share the same off-line stage and have different on-line stages as described below:

4.1 Off–line stage

In the off-line stage, snapshot solutions $\mathbf{q}_i^{(0)}(\boldsymbol{\omega}_n) = \mathbf{q}_i^{(0)}(\mu_{r,n})$ and $\mathbf{q}_i^{(1)}(\boldsymbol{\omega}_n)$, corresponding to the finite element solution coefficients for $\tilde{\boldsymbol{\theta}}_i^{(0,hp)}(\mu_{r,n})$ and $\boldsymbol{\theta}_i^{(1,hp)}(\boldsymbol{\omega}_n)$, are first obtained for a small number of sets of snapshot parameters $\boldsymbol{\omega}_n = [\omega_n, \mu_{r,n}]$, $n = 1, \ldots, N$, by solving systems of the form (6). Based on previous experience in [5], we choose the snapshot parameters $\boldsymbol{\omega}_n$ to be logarithmically spaced over a two dimensional grid. In addition, the solution snapshots $\mathbf{q}_i^{(0)}\left(\mu_{r,n}\right)$ were post-processed by applying a post-processing Poisson projection [15]:

$$\tilde{\boldsymbol{\theta}}_i^{(0,hp)} \to \nabla\Delta^{-1}\mathrm{div}\left(\tilde{\boldsymbol{\theta}}_i^{(0,hp)}\right), \tag{8}$$

which improves the gauging of $\tilde{\boldsymbol{\theta}}_i^{(0,hp)}$ without changing $\nabla \times \tilde{\boldsymbol{\theta}}_i^{(0,hp)}$. The matrices $\mathbf{D}_i^{(0)} \in \mathbb{R}^{N_{dof}\times N}$ and $\mathbf{D}_i^{(1)} \in \mathbb{C}^{N_{dof}\times N}$, $i = 1, 2, 3$, are then defined as a concatenation of $\mathbf{q}_i^{(0)}(\boldsymbol{\omega}_n)$ or $\mathbf{q}_i^{(1)}(\boldsymbol{\omega}_n)$ as

$$\mathbf{D}_i^{(s)} := \left[\mathbf{q}_i^{(s)}\left(\boldsymbol{\omega}_1\right), \mathbf{q}_i^{(s)}\left(\boldsymbol{\omega}_2\right), \mathbf{q}_i^{(s)}\left(\boldsymbol{\omega}_3\right), \ldots, \mathbf{q}_i^{(s)}\left(\boldsymbol{\omega}_N\right)\right]. \tag{9}$$

For the on-line stages of the PODP and POD–NN approaches, discussed in Sects. 4.2 and 4.3 respectively, the off-line stage continues with a singular value decomposition (SVD) applied to each $\mathbf{D}_i^{(s)}$ in order to extract modal information

$$\mathbf{D}_i^{(s)} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}} \approx \mathbf{U}^M\boldsymbol{\Sigma}^M\left(\mathbf{V}^M\right)^{\mathrm{H}}, \tag{10}$$

where we omit the dependence of i and (s) on the SVD matrices and their truncated counterparts for simplicity of presentation. In the above, H is the Hermitian and in the approximation $M < N$ corresponds to the level of truncation, which is determined by prescribing a tolerance TOL on the ordered relative singular values contained in the diagonal elements of $\boldsymbol{\Sigma}$. This truncation, results in a truncated matrix $\mathbf{U}^M \in \mathbb{C}^{N_{dof}\times M}$ containing the first M columns of the unitary matrix $\mathbf{U}$, a square matrix $\boldsymbol{\Sigma}^M \in \mathbb{R}^{M\times M}$ containing the truncated singular values on the diagonal and, $\mathbf{V}^M \in \mathbb{C}^{N\times M}$, which is obtained by taking the first M columns of the unitary matrix $\mathbf{V}$. Using (10), we recover an approximation to $\mathbf{q}_i^{(s)}(\boldsymbol{\omega}_n)$ as follows:

$$\mathbf{q}_i^{(s)}\left(\boldsymbol{\omega}_n\right) \approx \mathbf{U}^M\boldsymbol{\Sigma}^M\left(\left(\mathbf{V}^M\right)^{\mathrm{H}}\right)_{:,n}, \tag{11}$$

where $\left(\left(\mathbf{V}^M\right)^{\mathrm{H}}\right)_{:,n}$ refers to the nth column of $\left(\mathbf{V}^M\right)^{\mathrm{H}}$. Note that in the case of $s = 0$, the matrices $\mathbf{U}^M$ and $\mathbf{V}^M$ are real and so H can be replaced with transpose.

4.2 PODP - Projection based ROM

Following the PODP approach described by Wilson and Ledger in [5], we briefly describe how it can be extended to multiple parameters.

On–line stage In the on-line stage, we solve a small linear system (equation (24) [5]) of the form

$$\mathbf{A}^M(\boldsymbol{\omega})\mathbf{p}^M(\boldsymbol{\omega}) = \mathbf{r}^M(\boldsymbol{\omega}), \tag{12}$$

of size $M \times M$ for $\mathbf{p}^M(\boldsymbol{\omega}) \in \mathbb{C}^M$ where $\mathbf{A}^M(\boldsymbol{\omega}) := \left(\mathbf{U}^M\right)^{\mathrm{H}}\mathbf{A}(\boldsymbol{\omega})\mathbf{U}^M$, and $\mathbf{r}^M := \left(\mathbf{U}^M\right)^{\mathrm{H}}\mathbf{r}(\boldsymbol{\omega})$ (with reduction to real matrices for $s = 0$). Once $\mathbf{p}^M(\boldsymbol{\omega})$ is obtained, we use the approximation $\mathbf{q}_i^{(s)} \approx \mathbf{U}^M\mathbf{p}^M(\boldsymbol{\omega})$ [5]. By repeating this for $s = 0, 1$, $i = 1, 2, 3$ and combining with (7), allows us approximate $\tilde{\boldsymbol{\theta}}_i^{(0,hp)}$ and $\boldsymbol{\theta}_i^{(1,hp)}$ and, hence, the approximate $(\mathcal{M}^{\mathrm{PODP}})_{ij}$ for new $\boldsymbol{\omega}$.

4.2.1 Error Estimation

A detailed proof of an error estimate $\Delta(\boldsymbol{\omega})_{ij}$, which provides the upper bound

$$\left|(\mathcal{M}(\boldsymbol{\omega})^{\mathrm{PODP}})_{ij} - (\mathcal{M}(\boldsymbol{\omega}))_{ij}\right| \leq \Delta(\boldsymbol{\omega})_{ij}, \tag{13}$$

on the MPT coefficients obtained by the PODP approximation with respect to $(\mathcal{M}(\boldsymbol{\omega}))_{ij}$ obtained using the full-order finite element solve for $\boldsymbol{\omega} = [\omega]$ is established in [5]. This naturally extends to the case where $\boldsymbol{\omega} = [\omega, \mu_r]$ using

$$\begin{aligned}\Delta(\boldsymbol{\omega})_{ij} =& \frac{\alpha^3}{8\alpha_{LB}}\big(\|\hat{\boldsymbol{r}}_i(\boldsymbol{\omega})\|^2_{Y^{(hp)}} + \|\hat{\boldsymbol{r}}_j(\boldsymbol{\omega})\|^2_{Y^{(hp)}}\\ &+ \|\hat{\boldsymbol{r}}_i(\boldsymbol{\omega}) - \hat{\boldsymbol{r}}_j(\boldsymbol{\omega})\|^2_{Y^{(hp)}}\big),\end{aligned} \tag{14}$$

and, in this case, α_{LB} is the lower bound on a stability constant obtained by taking the smallest eigenvalue from an eigenvalue problem [11, pg 56] for the smallest frequency and inverse permeability of interest. $Y^{(hp)}$ is as defined in [5] and corresponds to the set of $\boldsymbol{H}(\mathrm{curl})$ conforming functions in the discretisation.

Similarly to [5], to efficiently compute $\hat{\boldsymbol{r}}_i(\boldsymbol{\omega})$ and $\hat{\boldsymbol{r}}_j(\boldsymbol{\omega})$, we build a discrete analogue of the residual by first constructing

$$\mathbf{G}^{(i,j)} = \left(\mathbf{W}^{(i)}\right)^{\mathrm{H}}\mathbf{M}_0^{-1}\mathbf{W}^{(j)}, \tag{15}$$

once for all $\boldsymbol{\omega}$ where $\mathbf{M}_0$ is the real symmetric mass matrix for the lowest order basis functions. To obtain $\mathbf{W}^{(i)}$ for $\boldsymbol{\omega} = [\omega, \mu_r]$, we consider the splittings

$$\mathbf{A} = \mathbf{B}^{(0)} + \mu_r^{-1}\mathbf{B}^{(1)} + \omega\mathbf{C}^{(1)}, \qquad \mathbf{r} = \omega\mathbf{r}^{(1)}, \tag{16}$$

where

$$\left(\mathbf{B}^{(0)}\right)_{ij} = \int_{\Omega\setminus\overline{B}} \nabla\times\mathbf{N}_i\cdot\nabla\times\mathbf{N}_j\,\mathrm{d}\boldsymbol{\xi} + \varepsilon\int_{\Omega\setminus\overline{B}}\mathbf{N}_i\cdot\mathbf{N}_j\,\mathrm{d}\boldsymbol{\xi}, \tag{17a}$$

$$\left(\mathbf{B}^{(1)}\right)_{ij} = \int_B \nabla\times\mathbf{N}_i\cdot\nabla\times\mathbf{N}_j\,\mathrm{d}\boldsymbol{\xi}, \tag{17b}$$

$$\left(\mathbf{C}^{(1)}\right)_{ij} = -\mathrm{i}\int_B \alpha^2\sigma_*\mu_0\mathbf{N}_i\cdot\mathbf{N}_j\,\mathrm{d}\boldsymbol{\xi}, \tag{17c}$$

and $\mathbf{r}^{(1)}$ is similarly constructed by taking out the factor ω from $\mathbf{r}$. Then, using $\mathbf{U}^{(M,i)}$ to denote $\mathbf{U}^M$ for the ith direction and $\boldsymbol{\theta}_i^{(1,hp)}$, and similarly for $\boldsymbol{\Sigma}^{(M,i)}$, we can determine $\mathbf{W}^{(i)}$ as

$$\mathbf{W}^{(i)} := \mathbf{P}_0^p\left(\mathbf{r}^{(1)},\ \mathbf{B}^{(0)}\mathbf{U}^{(M,i)},\ \mathbf{B}^{(1)}\mathbf{U}^{(M,i)},\ \mathbf{C}^{(1)}\mathbf{U}^{(M,i)}\right), \tag{18}$$

where $\mathbf{P}_0^p$ is the projection from order p to order 0 $\boldsymbol{H}(\mathrm{curl})$ conforming basis functions. It then follows that an efficient evaluation of the contributions to the error estimate are given by

$$\|\hat{\boldsymbol{r}}_i(\boldsymbol{\omega})\|^2_{Y^{(hp)}} = \left(\mathbf{w}^{(i)}(\boldsymbol{\omega})\right)^{\mathrm{H}}\mathbf{G}^{(i,i)}\left(\mathbf{w}^{(i)}(\boldsymbol{\omega})\right), \tag{19}$$

$$\begin{aligned}\|\hat{\boldsymbol{r}}_i(\boldsymbol{\omega}) - \hat{\boldsymbol{r}}_j(\boldsymbol{\omega})\|^2_{Y^{(hp)}} =& \|\hat{\boldsymbol{r}}_i(\boldsymbol{\omega})\|^2_{Y^{(hp)}} + \|\hat{\boldsymbol{r}}_j(\boldsymbol{\omega})\|^2_{Y^{(hp)}}\\ &-2\mathrm{Re}\left(\left(\mathbf{w}^{(i)}(\boldsymbol{\omega})\right)^{\mathrm{H}}\mathbf{G}^{(i,j)}\left(\mathbf{w}^{(j)}(\boldsymbol{\omega})\right)\right),\end{aligned} \tag{20}$$

for each parameter vector, $\boldsymbol{\omega}$, by updating

$$\mathbf{w}^{(i)}(\boldsymbol{\omega}) = \begin{bmatrix}\omega\\ -\mathbf{p}^{(M,i)}(\boldsymbol{\omega})\\ -\mu_r^{-1}\mathbf{p}^{(M,i)}(\boldsymbol{\omega})\\ -\omega\mathbf{p}^{(M,i)}(\boldsymbol{\omega})\end{bmatrix}. \tag{21}$$

In (18), $\mathbf{r}^{(1)}$ depends on μ_r, which means that, in our implementation, we construct the matrices iteratively, however, it is also possible to formulate an alternative construction of $\mathbf{W}^{(i)}$ and $\mathbf{w}^{(i)}(\boldsymbol{\omega})$ from (5), provided that the $\tilde{\boldsymbol{\theta}}_i^{(0)}$ and $\boldsymbol{\theta}_i^{(1)}$ problems are not split. The MPT coefficients in this case can be computed using the alternative (but equivalent) formulation in [2].

4.3 POD–NN - Neural Network Enhanced ROM

We follow an approach that is equivalent to the POD–NN approach described in [18, 19].

On–line stage In POD-NN, the approximate solution for new parameters $\boldsymbol{\omega}$ is taken as

$$\mathbf{q}_i^{(s)}(\boldsymbol{\omega}) \approx \mathbf{U}^M\boldsymbol{R}\left(\boldsymbol{\omega}, \boldsymbol{c}\left(\boldsymbol{\Sigma}^M\left(\mathbf{V}^M\right)^{\mathrm{H}}\right)\right), \tag{22}$$

and the mth component of $\boldsymbol{R}$, $R_m\left(\boldsymbol{\omega}, \boldsymbol{c}_m\right)$, is a prescribed function (e.g. a polynomial or some other smoothly varying differentiable function) whose coefficients $\boldsymbol{c}$ are found from solving the minimisation problem

$$\min_{\boldsymbol{c}}\frac{1}{N}\sum_{n=1}^{N}\left|\left(\boldsymbol{\Sigma}^M(\boldsymbol{V}^M)^{\mathrm{H}}\right)_{:,n} - \boldsymbol{R}\left(\boldsymbol{\omega}_n, \boldsymbol{c}\right)\right|^2 \tag{23}$$

Due to their superior interpolation properties, a neural network is used for this regression and, hence, the name POD–NN. Our implementation differs from [19] in that we train the network based on columns of $\boldsymbol{\Sigma}^M(\mathbf{V}^M)^{\mathrm{H}}$ rather than columns of $(\mathbf{U}^M)^{\mathrm{H}}\mathbf{D}$, but note that by (10) $\boldsymbol{\Sigma}^M(\mathbf{V}^M)^{\mathrm{H}} \approx (\mathbf{U}^M)^{\mathrm{H}}\mathbf{D}$ and so the two approaches are equivalent.

For the specific details of the implementation, which involves training individual networks for each direction $i = 1, 2, 3$ and $s = 0, 1$, we refer to [18]. Combining with (7), allows us approximate $\tilde{\boldsymbol{\theta}}_i^{(0,hp)}$ and $\boldsymbol{\theta}_i^{(1,hp)}$ and, hence, the approximate $(\mathcal{M}(\boldsymbol{\omega})^{\mathrm{POD-NN}})_{ij}$ for new $\boldsymbol{\omega}$.

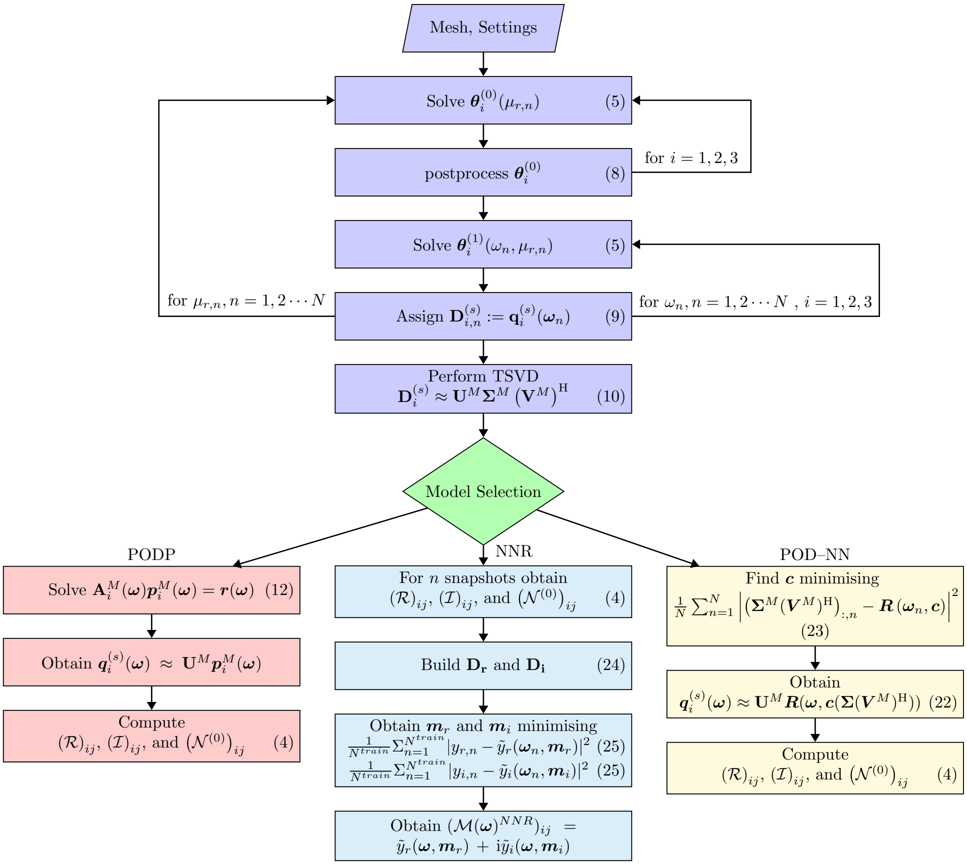

Mesh, Settings
Solve $\boldsymbol{\theta}_i^{(0)}(\mu_{r,n})$ (5)
postprocess $\boldsymbol{\theta}_i^{(0)}$ (8)
for $i = 1,2,3$
Solve $\boldsymbol{\theta}_i^{(1)}(\omega_n, \mu_{r,n})$ (5)
for $\mu_{r,n}, n = 1,2 \cdots N$
Assign $\mathbf{D}_{i,n}^{(s)} := \mathbf{q}_i^{(s)}(\boldsymbol{\omega}_n)$ (9)
for $\omega_n, n = 1,2 \cdots N$, $i = 1,2,3$
Perform TSVD
$\mathbf{D}_i^{(s)} \approx \mathbf{U}^M \mathbf{\Sigma}^M \left(\mathbf{V}^M\right)^{\mathrm{H}}$ (10)
Model Selection
PODP
NNR
POD–NN
Solve $\mathbf{A}_i^M(\boldsymbol{\omega})\boldsymbol{p}_i^M(\boldsymbol{\omega}) = \boldsymbol{r}(\boldsymbol{\omega})$ (12)
Obtain $\boldsymbol{q}_i^{(s)}(\boldsymbol{\omega}) \approx \mathbf{U}^M \boldsymbol{p}_i^M(\boldsymbol{\omega})$
Compute
$(\mathcal{R})_{ij}$, $(\mathcal{I})_{ij}$, and $\left(\mathcal{N}^{(0)}\right)_{ij}$ (4)
For n snapshots obtain
$(\mathcal{R})_{ij}$, $(\mathcal{I})_{ij}$, and $\left(\mathcal{N}^{(0)}\right)_{ij}$ (4)
Build $\mathbf{D_r}$ and $\mathbf{D_i}$ (24)
Obtain $\boldsymbol{m}_r$ and $\boldsymbol{m}_i$ minimising
$\frac{1}{N^{train}}\Sigma_{n=1}^{N^{train}}|y_{r,n} - \tilde{y}_r(\boldsymbol{\omega}_n, \boldsymbol{m}_r)|^2$ (25)
$\frac{1}{N^{train}}\Sigma_{n=1}^{N^{train}}|y_{i,n} - \tilde{y}_i(\boldsymbol{\omega}_n, \boldsymbol{m}_i)|^2$ (25)
Obtain $(\mathcal{M}(\boldsymbol{\omega})^{NNR})_{ij} =$
$\tilde{y}_r(\boldsymbol{\omega}, \boldsymbol{m}_r) + \mathrm{i}\tilde{y}_i(\boldsymbol{\omega}, \boldsymbol{m}_i)$
Find $\boldsymbol{c}$ minimising
$\frac{1}{N}\sum_{n=1}^{N}\left|\left(\mathbf{\Sigma}^M(\boldsymbol{V}^M)^{\mathrm{H}}\right)_{:,n} - \boldsymbol{R}(\boldsymbol{\omega}_n, \boldsymbol{c})\right|^2$
(23)
Obtain
$\boldsymbol{q}_i^{(s)}(\boldsymbol{\omega}) \approx \mathbf{U}^M \boldsymbol{R}(\boldsymbol{\omega}, \boldsymbol{c}(\mathbf{\Sigma}(\boldsymbol{V}^M)^{\mathrm{H}}))$ (22)
Compute
$(\mathcal{R})_{ij}$, $(\mathcal{I})_{ij}$, and $\left(\mathcal{N}^{(0)}\right)_{ij}$ (4)

Fig. 2 Flowchart summarising the PODP, POD–NN, and NNR methods for quickly computing the MPT coefficients

4.4 NNR - Neural Network Regression of MPT coefficients

As a final alternative, we consider a neural network regression technique for predicting $(\mathcal{M}(\boldsymbol{\omega})^{\mathrm{NNR}})_{ij}$ for new parameters $\boldsymbol{\omega}$. This is the least invasive of the techniques considered as the prediction can be obtained by curve-fitting. However, it lacks the physical insights that are gained from using the modal information in the other POD-based approaches and so we expect it to be less accurate for the same parameter snapshots. To fix ideas, let $(\mathcal{M}(\boldsymbol{\omega}_n))_{ij} = y_{\mathrm{r},n} + \mathrm{i}y_{\mathrm{i},n}$ and introduce the dictionaries of ordered pairs

$$\mathbf{D}_{\mathrm{r}} := \left\{ \left(\boldsymbol{\omega}_1, y_{\mathrm{r},1}\right), \left(\boldsymbol{\omega}_2, y_{\mathrm{r},2}\right), \dots, \left(\boldsymbol{\omega}_N, y_{\mathrm{r},N}\right) \right\}, \tag{24a}$$

$$\mathbf{D}_{\mathrm{i}} := \left\{ \left(\boldsymbol{\omega}_1, y_{\mathrm{i},1}\right), \left(\boldsymbol{\omega}_2, y_{\mathrm{i},2}\right), \dots, \left(\boldsymbol{\omega}_N, y_{\mathrm{i},N}\right) \right\}. \tag{24b}$$

By splitting each of the dictionaries into training and testing parts using a 15% reserve such that $N^{train} < N$, we then apply feed-forward networks [22] that are trained to minimise the functionals

$$\begin{aligned} &\min_{\boldsymbol{m}_{\mathrm{r}}} \frac{1}{N^{train}} \sum_{n=1}^{N^{train}} |y_{\mathrm{r},n} - \tilde{y}_{\mathrm{r}}(\boldsymbol{\omega}_n, \boldsymbol{m}_{\mathrm{r}})|^2, \\ &\min_{\boldsymbol{m}_{\mathrm{i}}} \frac{1}{N^{train}} \sum_{n=1}^{N^{train}} |y_{\mathrm{i},n} - \tilde{y}_{\mathrm{i}}(\boldsymbol{\omega}_n, \boldsymbol{m}_{\mathrm{i}})|^2 \end{aligned} \tag{25}$$

for the model parameters $\boldsymbol{m}_{\mathrm{r}}$ and $\boldsymbol{m}_{\mathrm{i}}$ in order to predict $(\mathcal{M}(\boldsymbol{\omega})^{NNR})_{ij} = \tilde{y}_r(\boldsymbol{\omega}, \boldsymbol{m}_{\mathrm{r}}) + \mathrm{i}\tilde{y}_{\mathrm{i}}(\boldsymbol{\omega}, \boldsymbol{m}_{\mathrm{i}})$. This can equivalently be used to directly build a regression of the MPT eigenvalues.

Fig. 2 shows a flowchart summarising the 3 different methods for computing the MPT coefficients with associated equation numbers.

4.5 Software

Our practical implementations[1] build on the `MPT-Calculator` software initially developed by Wilson and Ledger [5], which uses the `NGSolve` library (version 6.2.2204) for the finite element computations [12–15]. For our neural-network computations, we use the `Scikit-Learn` library

[1] The github repository https://github.com/MPT-Calculator/MPT-Calculator is publicly available.

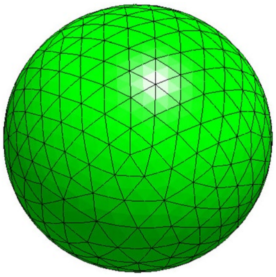


Fig. 3 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s: Surface mesh of B using an unstructured tetrahedral mesh with size $h = 0.2$

(version 1.1.2) and consider the tanh and sigmoid activation functions. Prior to training, we scale the training data so that it has mean 0 and standard deviation 1. We use a quasi-Newton type limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) method [23–25] to determine the model parameters $\boldsymbol{m}_\mathrm{r}$ and $\boldsymbol{m}_\mathrm{i}$. To determine the hyperparameters, for the feed-forward network we employ a grid-based search. We consider $\ell = 1, 2$, and 3 hidden layers, each with either $t = 2, 4, 8, 16, 32$, or 64 neurons and either the tanh or logistic activation functions.

5 Numerical Examples

In this section, we first compare the PODP, POD–NN and NNR approaches for the MPT characterisation of a conducting permeable sphere for different ω and μ_r and then show the predictive capability of the approach for a geometry that does not have an analytical solution.

5.1 Conducing Permeable Sphere

This subsection discusses the case where B_α is a conducting permeable sphere with radius $\alpha = 0.01$ m, conductivity $\sigma_* = 10^6$ S/m, relative permeability $1 \le \mu_r \le 50$ and exciting frequency $10^1 \le \omega \le 10^5$ rad/s. An analytical solution is available for $(\mathcal{M})_{ij}$ for this geometry [26] in the form $(\mathcal{M})_{ij} = m(\alpha B, \omega, \sigma_*, \mu_r)\delta_{ij}$, which shows that $\mathcal{M}$ is a multiple of identity in this case and, hence, $\tilde{\boldsymbol{\theta}}_1^{(0)} = \tilde{\boldsymbol{\theta}}_2^{(0)} = \tilde{\boldsymbol{\theta}}_3^{(0)}$ and $\boldsymbol{\theta}_1^{(1)} = \boldsymbol{\theta}_2^{(1)} = \boldsymbol{\theta}_3^{(1)}$ for the continuous problem.

To compute snapshot solutions for this geometry, we consider B to be a unit radius sphere centred at the origin and set up a finite computational domain Ω in the form of a sphere of radius 200 units. The domain is discretised by a quasi-uniform tetrahedral mesh of element size 0.2 units and 57 698 elements as illustrated in Fig. 3. The curved surface of Γ is approximated by 5th order polynomials.

5.1.1 Full Order Model Solutions

Before considering the accuracy of the reduced order model approaches, we first consider the accuracy of the full-order model for computing $(\mathcal{M}^{hp})_{ij}$ using different uniform polynomial orders $p = 0, 1, 2, 3, 4$ for $\tilde{\boldsymbol{\theta}}_i^{(0,hp)}$ and $\boldsymbol{\theta}_i^{(1,hp)}$. In Fig. 4, we show the convergence of the first eigenvalues $\lambda_1(\tilde{\mathcal{R}}^{hp})$ and $\lambda_1(\mathcal{I}^{hp})$ obtained using the full order model to $\lambda_1(\tilde{\mathcal{R}})$ and $\lambda_1(\mathcal{I})$ for $10 \le \omega \le 10^5$ rad/s and $\mu_r = 1, 20$. We observe a rapid convergence of eigenvalues to the exact solution. The convergence of the other eigenvalues is similar.

While the solutions obtained from p-refinement are accurate, for each choice of parameters $\boldsymbol{\omega} = [\omega, \mu_r]$ a preconditioned conjugate gradient solver is applied to solve (6), which requires repeated matrix–vector products involving the sparse matrix $\mathbf{A}$ with nz non-zero entries. Repeated solution of the linear system (6) for a large number of evaluation parameters is expensive.

5.1.2 Reduced Order Model Solutions

Off–line stage

Unless otherwise stated, we consider $N = 16^2$ snapshot full order model solutions using the aforementioned discretisation and uniform order $p = 2$ elements. This is then evaluated over a $K = 32^2$ grid of evaluation parameters. This was identified by considering the minimum number of snapshots for which the PODP, POD–NN and NNR all gave reliable results with a relative root mean squared error (rRMSE) $e \le 10^{-2}$ where e is

$$e = \frac{\sqrt{\sum_{k=1}^{K} |z^{\mathrm{APP}}(\boldsymbol{\omega}_k) - z^{hp}(\boldsymbol{\omega}_k)|^2}}{\sqrt{\sum_{k=1}^{K} |z^{hp}(\boldsymbol{\omega}_k)|^2}} \tag{26}$$

Denoting $z^{hp}(\boldsymbol{\omega}) := \lambda_1(\tilde{\mathcal{R}}(\boldsymbol{\omega})^{hp}) + \mathrm{i}\lambda_1(\mathcal{I}(\boldsymbol{\omega})^{hp})$ and $z^{\mathrm{APP}}(\boldsymbol{\omega}) := \lambda_1(\tilde{\mathcal{R}}(\boldsymbol{\omega})^{\mathrm{APP}}) + \mathrm{i}\lambda_1(\mathcal{I}(\boldsymbol{\omega})^{\mathrm{APP}})$ for $k = 1, 2, \cdots, K$ evaluations parameters, where APP is used to denote either PODP, POD–NN or NNR.

The snapshots are computed for logarithmically spaced parameters $\omega_n = 10^{\tilde{\omega}_n}$, where $\tilde{\omega}_n$ is drawn from a 16 sample linearly spaced distribution between 1 and 5, and $\mu_{r,n} = b^{\tilde{\mu}_{rn}}$, where $\tilde{\mu}_{rn}$ is obtained from the linear distribution between 0 and $\log_b(50)$. The base b is chosen to be $50^{1/5}$.

Then, once (9) and (10) have been applied, we can extract the modal information from the SVD. In the case of $\mathbf{D}_i^{(0)}$, we obtain the decay of the singular values shown in Fig. 5 where the improvement in the decay of the singular values by including (8) is clear. Although not shown, in the case of $\mathbf{D}_i^{(1)}$, the projection (8) is not appropriate, since the projection would remove the gradients fields needed inside the object in this case. Henceforth, we select $TOL = 10^{-6}$ for the truncated SVDs (TSVDs) of $\mathbf{D}_i^{(0)}$ and $\mathbf{D}_i^{(1)}$. This corresponds to $M = 3$ modes per $\tilde{\boldsymbol{\theta}}_i^{(0)}$ and $M = 20$ modes per $\boldsymbol{\theta}_i^{(1)}$.

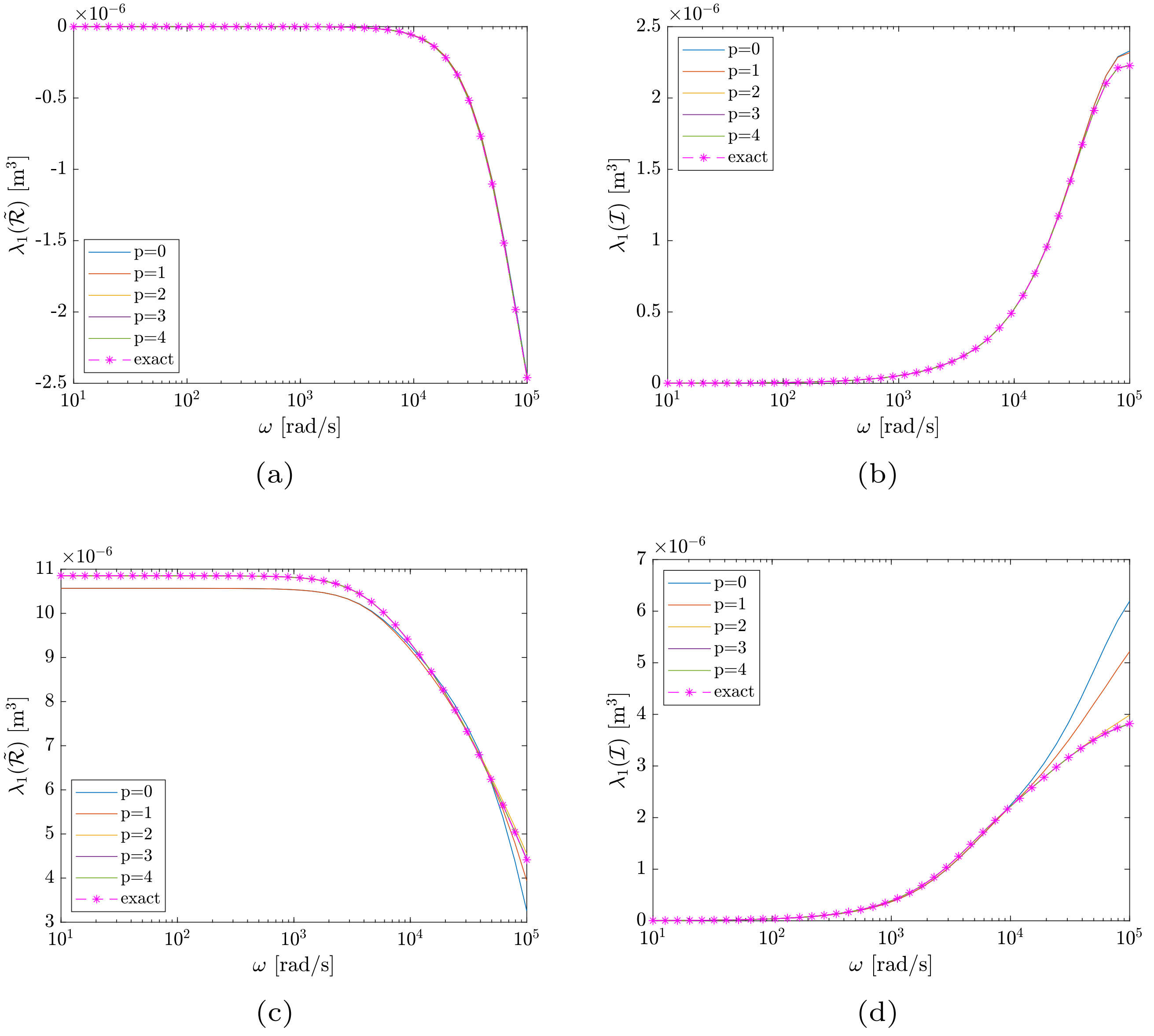


Fig. 4 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m and $10^1 \le \omega \le 10^5$ rad/s: p–convergence of the full-order model solutions showing (a) $\lambda_1(\tilde{\mathcal{R}}^{hp})$ for $\mu_r = 1$, (b) $\lambda_1(\mathcal{I}^{hp})$ for $\mu_r = 1$, (c) $\lambda_1(\tilde{\mathcal{R}}^{hp})$ for $\mu_r = 20$, and (d) $\lambda_1(\mathcal{I}^{hp})$ for $\mu_r = 20$

Online stage–PODP

Applying the approach described in Sect. 4.2 leads to the results shown in Fig. 6 for $\lambda_1(\tilde{\mathcal{R}}^{\text{PODP}})$, and $\lambda_1(\mathcal{I}^{\text{PODP}})$ for the case of $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s. This figure shows that the reduced order model is in excellent agreement with the full order model solutions at the snapshot values and the prediction for other parameters follows the expected trends. The behaviour of the other eigenvalues is similar.

In order to certify the PODP method, the approach from Sect. 4.2.1 is applied leading to the results for $(\tilde{\mathcal{R}}^{\text{PODP}} \pm \Delta)$ shown in Fig. 7 for the cases where $N = 6^2$ and $N = 16^2$. In this figure, we observe that the certification reduces to the full order model solutions at the snapshot values and shows that PODP is highly reliable for a large range of ω and μ_r values in both cases. The larger error bounds for large ω and μ_r using $N = 6^2$ indicates that the solution is less reliable in this case, although the comparison with the snapshot values shows it is accurate. We emphasise that, despite the effectivity index of the upper bound being large in this region, this certification can be computed at only a small additional cost during the online stage of the ROM and, hence, stills provides useful information to assess our confidence in the PODP prediction. The confidence in the prediction can be improved by increasing from $N = 6^2$ to $N = 16^2$ as the figure

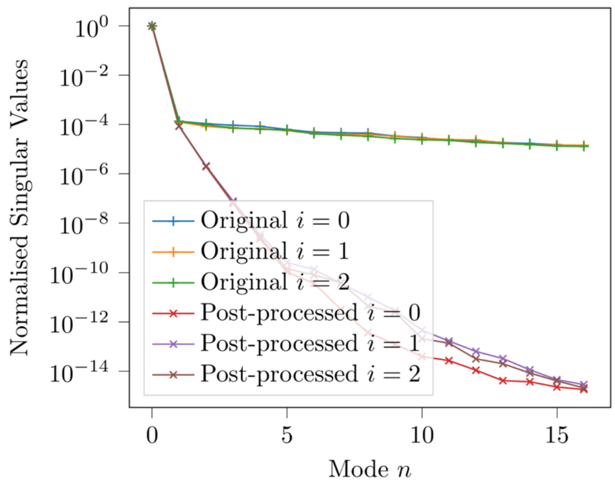


Fig. 5 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s: Comparison of $(\boldsymbol{\Sigma}^M)_{mm}/(\boldsymbol{\Sigma}^M)_{11}$ with and without the application of (8) for $\mathbf{D}_i^{(0)}$

shows. Alternatively, we can improve the confidence in the prediction, by just adding additional snapshots corresponding to the locations where $(\Delta)_{ij}$ is large. Similar behaviour is observed for $(\mathcal{I}^{\mathrm{PODP}} \pm \Delta)$ and for the other eigenvalues for this problem. It was also observed when the PODP approach was applied for a single parameter in [5].

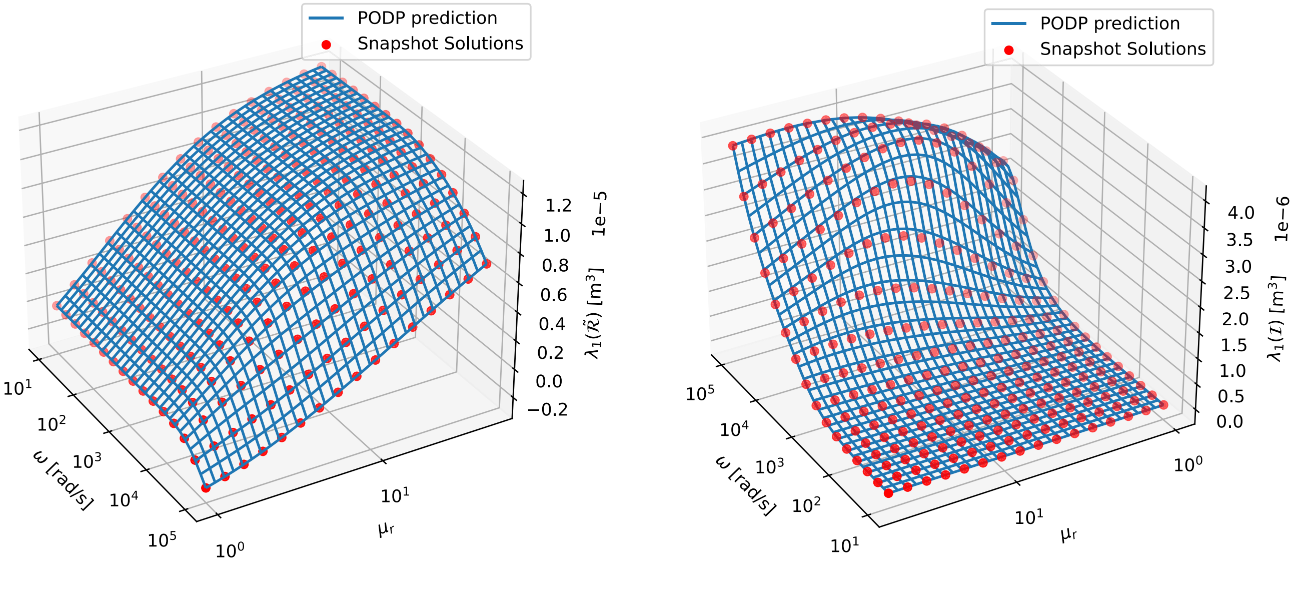
Online stage POD–NN

In this section, we consider the results obtained by applying the POD–NN described in Sect. 4.3. The neural networks used for POD–NN were obtained using a cross-validated grid search method resulting in a choice of 2 hidden layers, each with 8 neurons and a tanh activation function. We also observed that there was no significant change in accuracy when including a L_2 regularisation term in the training of the network and performance was degraded when considering more than $N = 16^2$ snapshots. Of additional note, the performance of the neural network was found to be stable, satisfying a training tolerance of 10^{-10} over at least $2 \le \ell \le 3$ layers and $2^3 \le t \le 2^6$ neurons.

For our implementation of POD–NN, a new network is trained and evaluated for each direction, $s = 0, 1$ and i with the real and imaginary parts of $\mathbf{D}_i^{(s)}$ being concatenated so that 6 training sets with M ordered pairs $(\mathbf{x}^{(m)}, \mathbf{y}^{(m)})$, where $\mathbf{x}^{(m)} \in \mathbb{R}^2$ and $\mathbf{y}^{(m)} \in \mathbb{R}^{2N}$, are formed. These training sets are used to train 6 different networks. Once this has been completed, the predictions shown in Fig. 8 for $\lambda_1(\tilde{\mathcal{R}}^{\mathrm{POD-NN}})$ and $\lambda_1(\mathcal{I}^{\mathrm{POD-NN}})$ for the case of $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s are obtained. Like the PODP case, the results show that this ROM is also in excellent agreement with the full-order model solutions at the snapshot values and the prediction for other parameters follows the expected trends. The behaviour for the other eigenvalues is similar.

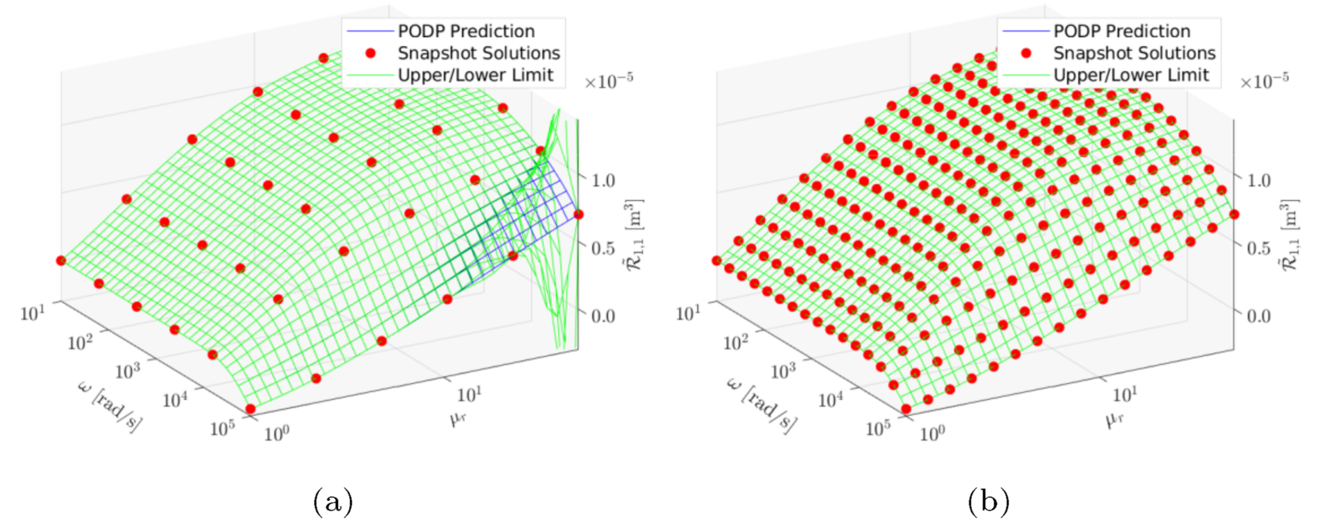
NNR regression

For simplicity of presentation, and ease of comparison, we consider the same neural network architecture that was previously used for the POD–NN scheme for NNR.



(a) (b)

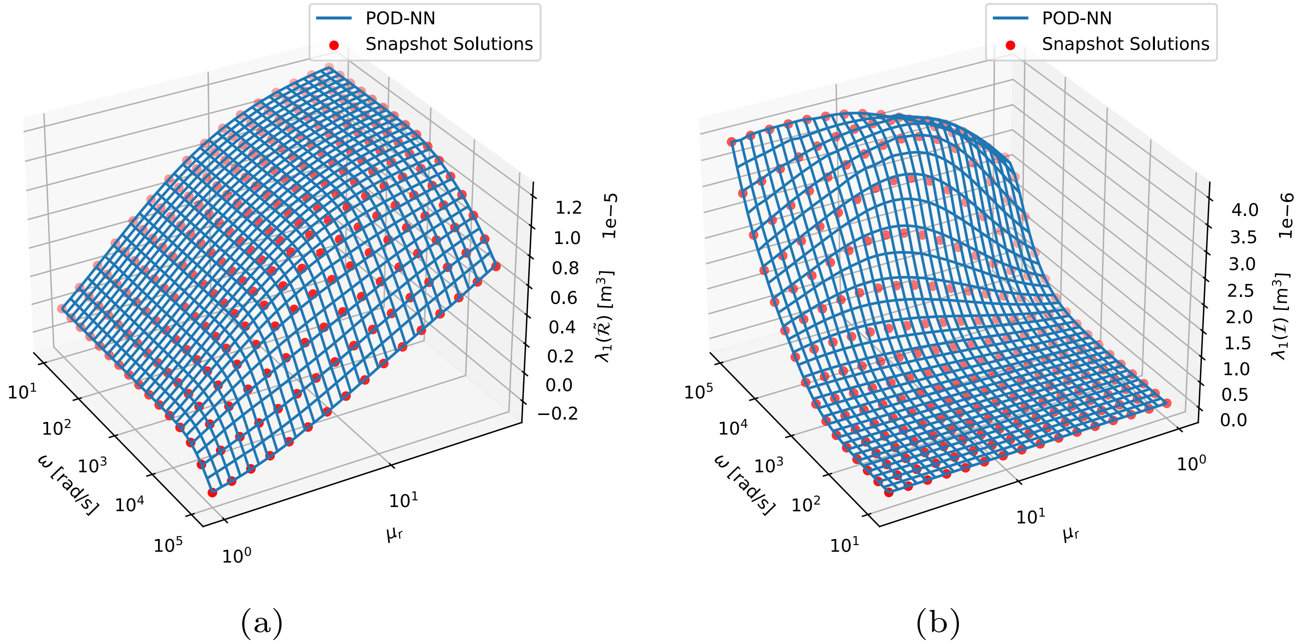
Fig. 6 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s: Results for (a) $\lambda_1(\tilde{\mathcal{R}}^{\mathrm{PODP}})$, and (b) $\lambda_1(\mathcal{I}^{\mathrm{PODP}})$ using $N = 16^2$ and $K = 32^2$. Note, (b) has been rotated in this, and subsequent, surface plots of the imaginary eigenvalues. In this example, the rRMSE is $e = 3.0 \times 10^{-7}$



(a)

(b)

Fig. 7 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \leq \mu_r \leq 50$ and $10^1 \leq \omega \leq 10^5$ rad/s: Results for $(\tilde{\mathcal{R}}^{\mathrm{PODP}} \pm \Delta)_{1,1}$ obtained using (a) $N = 6^2$ and (b) $N = 16^2$



(a)

(b)

Fig. 8 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \leq \mu_r \leq 50$ and $10^1 \leq \omega \leq 10^5$ rad/s: Results for (a) $\lambda_1(\tilde{\mathcal{R}}^{\mathrm{POD-NN}})$, and (b) $\lambda_1(\mathcal{I}^{\mathrm{POD-NN}})$ using $N = 16^2$ and $K = 32^2$. In this example, the rRMSE is $e = 2.0 \times 10^{-3}$

Applying the approach from Sect. 4.4 leads to the results for $\lambda_1(\tilde{\mathcal{R}}^{\mathrm{NNR}})$, and $\lambda_1(\mathcal{I}^{\mathrm{NNR}})$ for the case of $1 \leq \mu_r \leq 50$ and $10^1 \leq \omega \leq 10^5$ rad/s shown in Fig. 9. As with the PODP and POD–NN methods, the NNR method shows good visual agreement with the snapshot values and the prediction follows the expected trends. In addition, evaluating the rRMSE between the model and the testing samples gives $e < 0.01$. The results for the other eigenvalues are similar.

5.1.3 Methodology Comparison

Our focus in Sect. 5.1.2 was on a relatively fine off–line stage using $N = 16^2$ snapshot solutions where it was observed that the PODP, POD–NN and NNR approaches

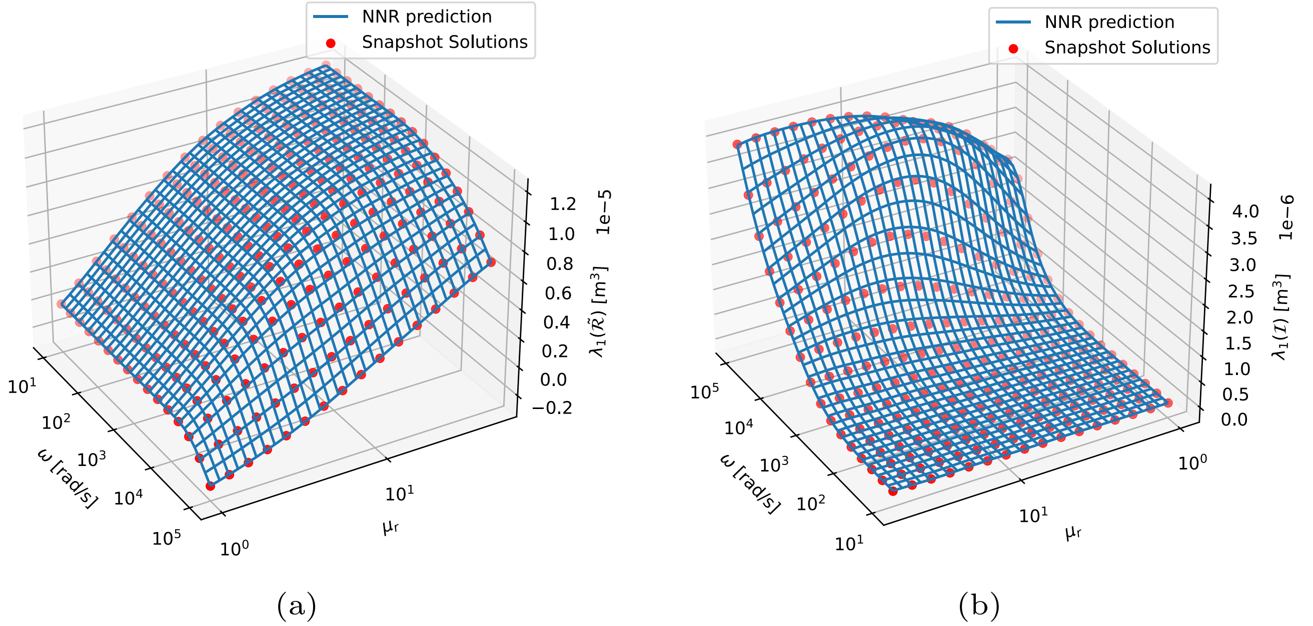


Fig. 9 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \leq \mu_r \leq 50$ and $10^1 \leq \omega \leq 10^5$ rad/s: Results for (a) $\lambda_1(\tilde{\mathcal{R}}^{\mathrm{NNR}})$, and (b) $\lambda_1(\mathcal{I}^{\mathrm{NNR}})$ using $N = 16^2$ and $K = 32^2$. In this example, the rRMSE is $e = 5.1 \times 10^{-3}$

all produced accurate results for $\lambda_1(\tilde{\mathcal{R}})$ and $\lambda_1(\mathcal{I})$ when evaluated for parameters $1 \leq \mu_r \leq 50$ and $10^1 \leq \omega \leq 10^5$ rad/s. We now wish to examine the performance of each method by evaluating the relative error using

We choose to evaluate e over a grid of $K = 32 \times 32$ points that are different from the snapshot parameters. The snapshots are generated using logarithmically spaced parameters corresponding to $N = (2)^2, (2^2)^2, (2^3)^2, (2^4)^2 = 4, 16, 64, 256$ snapshots and e evaluated at combinations of μ_r and ω corresponding to a 32 by 32 grid over the range $10^1 \leq \omega \leq 10^5$ rad/s and $1 \leq \mu_r \leq 50$.

Figure 10 shows e when evaluated for $1 \leq \mu_r \leq 50$, $10^1 \leq \omega \leq 10^5$ rad/s in a 32 by 32 grid as a function of N for the different methods. We can see that the PODP scheme performs significantly better than the POD-NN and NNR interpolants and, in particular, the PODP scheme leads to an approximation that is more than 4 orders of magnitude more accurate. Nonetheless, if $N \geq (8)^2$ snapshots are used, POD–NN leads to a solution with error $e \leq 0.01$. The behaviour for other combinations of μ_r and ω is similar. During the training process, a cross-validated grid search approach was used to find optimal hyperparameters. Nevertheless, better performance may be obtained by a smaller tolerance, performing an optimisation over a wider range of possible hyperparameters, or changing the activation function.

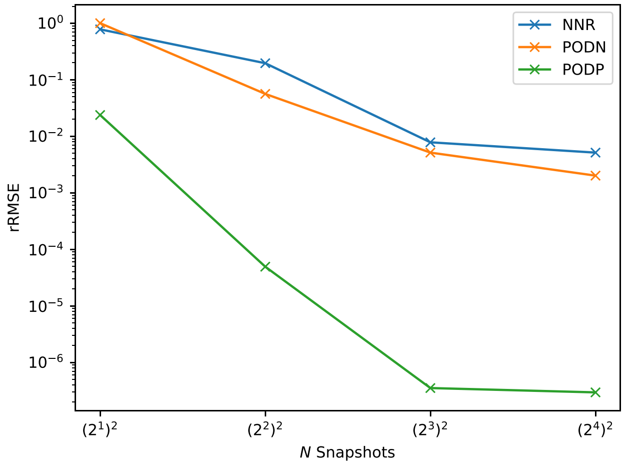
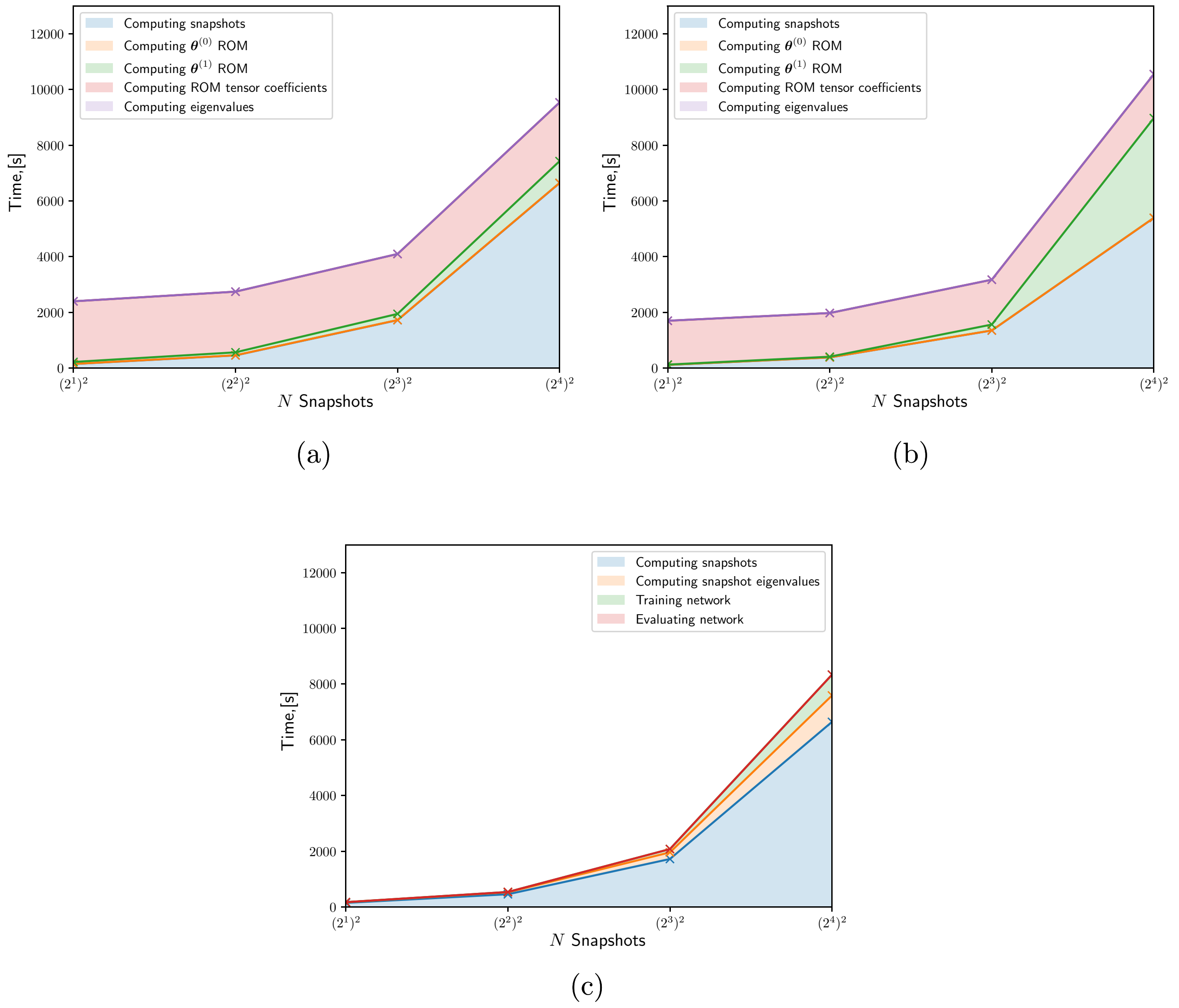


Fig. 10 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \leq \mu_r \leq 50$ and $10^1 \leq \omega \leq 10^5$ rad/s: Results for e obtained for a 32×32 grid of evaluation parameters over the range $1 \leq \mu_r \leq 50$, $10^1 \leq \omega \leq 10^5$ rad/s for different N and the PODP, POD–NN, NNR methods

Next, we consider a comparison between the computational time using a sequential methodology for the different methods in Fig. 11. We show the wall clock time taken to compute the snapshot solutions corresponding to $\mathbf{q}\left(\boldsymbol{\omega}_n\right)$ and present the timings required to obtain $\lambda_1(\tilde{\mathcal{R}}(\boldsymbol{\omega})^{\mathrm{APP}})$ and $\lambda_1(\mathcal{I}(\boldsymbol{\omega})^{\mathrm{APP}})$ at $K = 32^2$ different choices of $\boldsymbol{\omega}$ corresponding to the aforementioned evaluation values of ω and μ_r. These timings include the time required to optimise the hyperparameters using a cross validated grid-based search considering $\ell = 1, 2$ or 3 layer networks with $t = 1, 2, 4, 8, 16,$



(a)

(b)

(c)

Fig. 11 Conducting permeable sphere with $\alpha = 0.01$ m, $\sigma_* = 10^6$ S/m, $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s: Comparison of computation times for different N for the (a) PODP, (b) POD–NN, and (c) NNR methods

or 64 neurons. We also consider logistic or tanh activation functions and a small regularisation term of $0, 10^{-5}$ or 10^{-7}. A more expansive set of hyperparameters, a smaller training tolerance, or a different search strategy may result in a significantly increased time.

The figure shows a a significant acceleration in computational time for the NNR method, which is due to the need to only compute $\lambda_1(\tilde{\mathcal{R}}(\boldsymbol{\omega}_n)^{hp})$ and $\lambda_1(\mathcal{I}(\boldsymbol{\omega}_n)^{hp})$ for each set of snapshot parameters $\boldsymbol{\omega}_n$, whereas the PODP and POD–NN methods require the evaluation of $\lambda_1(\tilde{\mathcal{R}}(\boldsymbol{\omega})^{\text{APP}})$ and $\lambda_1(\mathcal{I}(\boldsymbol{\omega})^{\text{APP}})$ at the evaluation points $\boldsymbol{\omega}$. In each case, the PODP, POD–NN, and NNR methods perform significantly faster than the corresponding full order solution for these evaluation parameters which takes a wall clock time of 45965 s (approximately 13 h). PODP and POD–NN both share the same off–line stage and construction of the reduced order model. A detailed description for computational costs associated with POD is provided in [11, pg. 21-29]. Training the POD–NN and NNR neural networks depends heavily on the network architecture and tolerances but typically relies on efficient quasi-Newton optimisations. The LBFGS, which we have used here, is well suited to large dimensional problems given its linear computational cost. See [27, pg. 224-233].

Timings were performed using a 6 core Intel i5 10600 CPU and 64 GB of RAM where the multiple cores of this machine were used computing the snapshots in parallel, but not for the timings in Fig. 11. From the figures, we see that computing and postprocessing the snapshots constitutes a majority of the computation time and reducing this cost is a key benefit of the NNR method. Furthermore, training (including the optimisation to find the hyperparameters) and evaluating the ROM and neural networks is extremely quick for the relatively shallow neural networks considered.

From Figs. 10 and 11, we see that the PODP method is significantly more accurate than the NNR and POD–NN methods, and of a similar computational expense to the POD–NN method, however, this is at the expense of significant code intervention. For each method, many of the calculations are trivially parallelisable and a parallel implementation using a machine with multiple cores can be used to reduce the computational cost. In the case of NNR, which is the least invasive to implement, a parallel implementation of the calculation of $\lambda_1(\tilde{\mathcal{R}}(\boldsymbol{\omega}_n)^{hp})$ and $\lambda_1(\mathcal{I}(\boldsymbol{\omega}_n)^{hp})$ at all snapshots $\boldsymbol{\omega}_n$ would reduce the time, but, given that the largest contribution to the processing time is the computation of the snapshot solutions, which is shared across all three methods, additional parallelism is less effective than the PODP and POD–NN cases. In the case of PODP, which is the most invasive to implement, we see it has the smallest growth in computational time for this example, with solving the smaller linear system being quick and post-processing taking a constant time. However, if the size M of the ROM needed becomes large, the memory usage and computational time of this approach may increase significantly. The POD–NN method does not require access to $\mathbf{A}$ and $\mathbf{r}$ and is ideally suited to closed code bases.

5.2 Hammer Head Example

A metal claw head hammer is considered as a more complex example, modelled as 440A stainless steel $\sigma_* = 1.7 \times 10^6$ S/m [28, pg 894], and the non–dimensional object B is chosen to be such that $\alpha = 0.001$ m. Similarly to the sphere example, the object B for this case is surrounded by a large non-conducting region $[-1000, 1000]^3$ units and consists of 51 095 unstructured tetrahedral elements. The maximum mesh size inside the object is $h = 5$ units and the distribution of the element on the surface of B is illustrated in Fig. 12. In practice, many magnetic materials, such as steel, have a non–linear $\boldsymbol{B} - \boldsymbol{H}$ constitutive behaviour and, thus, μ_r depends on $\boldsymbol{H}$ and may vary by several orders of magnitude, although, for field strengths involved in metal detection, the linear relationship $\boldsymbol{B} = \mu_r \mu_0 \boldsymbol{H}$ typically holds. Nonetheless, it not always straightforward to find the correct μ_r for the characterisation. Using the ROM with two parameters allows us to explore the effects of increased μ_r at reduced computation time.

Given the success of the PODP approach in the previous section, we also apply this approach for the hammer-head example.

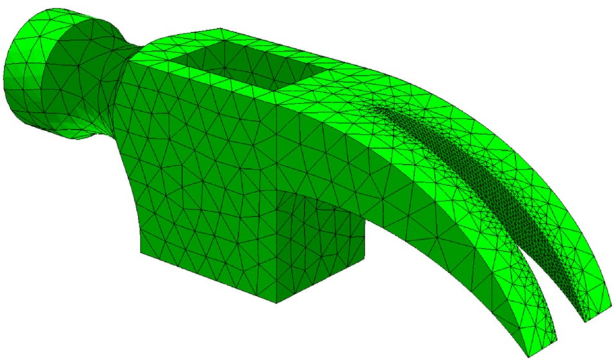


Fig. 12 Conducting permeable hammer head with $\alpha = 0.001$ m, $\sigma_* = 1.7 \times 10^6$ S/m, $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s: Surface mesh of B using an unstructured tetrahedral mesh with maximum size $h = 5$

Off–line solutions

By performing a p–convergence study on the mesh of 51 095 unstructured tetrahedral elements we found that using uniform order $p = 2$ order elements are sufficient to obtain converged results for $(\mathcal{M})_{ij}$ at the snapshot parameters. Then, using this discretisation, $N = 16^2$ full order model solution snapshots were generated corresponding to logarithmically spaced snapshot parameters, in a similar way as described in Sect. 5.1.2. Similarly to Sect. 5.1.2, TOL was set at 10^{-6} for both $\tilde{\boldsymbol{\theta}}^{(0)}$ and $\boldsymbol{\theta}^{(1)}$ problems.

On–line solutions

Applying the approach described in Sect. 4.2 leads to the results shown in Fig. 13 for $\lambda_1(\tilde{\mathcal{R}}^{\text{PODP}})$, and $\lambda_1(\mathcal{I}^{\text{PODP}})$ for the case of $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s. This figure shows that the ROM is in excellent agreement with the full-order model solutions at the snapshot values and the prediction for other parameters follows the expected trends. The behaviour for the other eigenvalues is similar.

Similarly, the estimated error certificates for this object shows that the ROM and full-order solutions are in good agreement, as illustrated by Fig 14 where $(\tilde{\mathcal{R}} \pm \Delta)_{1,1}$ and $(\mathcal{I} \pm \Delta)_{1,1}$ are shown. Similar performance is observed for the other coefficients.

6 Conclusion

In this article, we discuss alternative approaches to efficiently computing the complex MPT coefficients for different objects under a two-dimensional range of material properties. We have extended the PODP ROM discussed in [5] to two dimensions, and compared it with less invasive neural network-based regression techniques POD-NN and NNR.

A series of numerical examples are provided for the cases of a conducting permeable sphere, and a hammer-head modelled as 440 stainless steel over a wide frequency

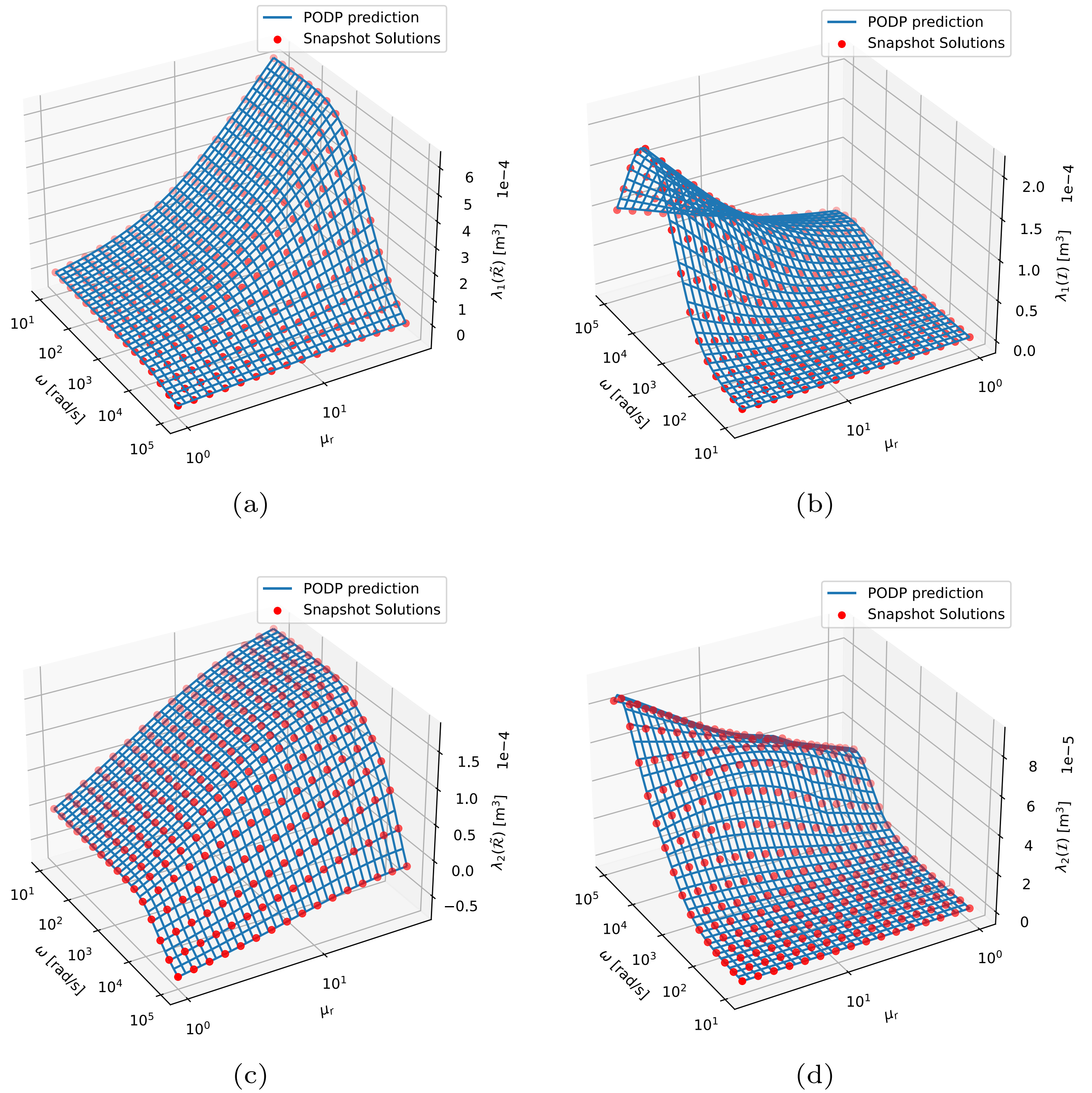


Fig. 13 Conducting permeable hammer head with $\alpha = 0.001$ m, $\sigma_* = 1.7 \times 10^6$ S/m, $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s: Results for (a) $\lambda_1(\tilde{\mathcal{R}}^{\text{PODP}})$, (b) $\lambda_1(\mathcal{I}^{\text{PODP}})$, (c) $\lambda_2(\tilde{\mathcal{R}}^{\text{PODP}})$, and (d) $\lambda_2(\mathcal{I}^{\text{PODP}})$ using $N = 16^2$. The rRMSE for this example is $e = 1.96 \times 10^{-6}$

range and considering a range of permeabilities. Our results have shown that the PODP method performs most accurately, however, if such high accuracy is not required, then the POD–NN method provides a less intrusive alternative. While the on-line stage of POD-NN is quick, the training and optmisation of hyper-parameters can add significant costs to the off-line stage depending on the optimisation strategy and architectures of the neural networks considered. Due to the inclusion of a more effective gauging for the $\boldsymbol{\theta}^{(0)}$ problem, we achieve faster decay of the singular values, and a more accurate ROM than the one presented in [5].

Future work involves applying the presented approaches to generate a large dictionary of MPT spectral signatures for different magnetic objects and to improving the methodology for resolving the fields in the thin skin depths for objects with very hight μ_r. In addition, it would be

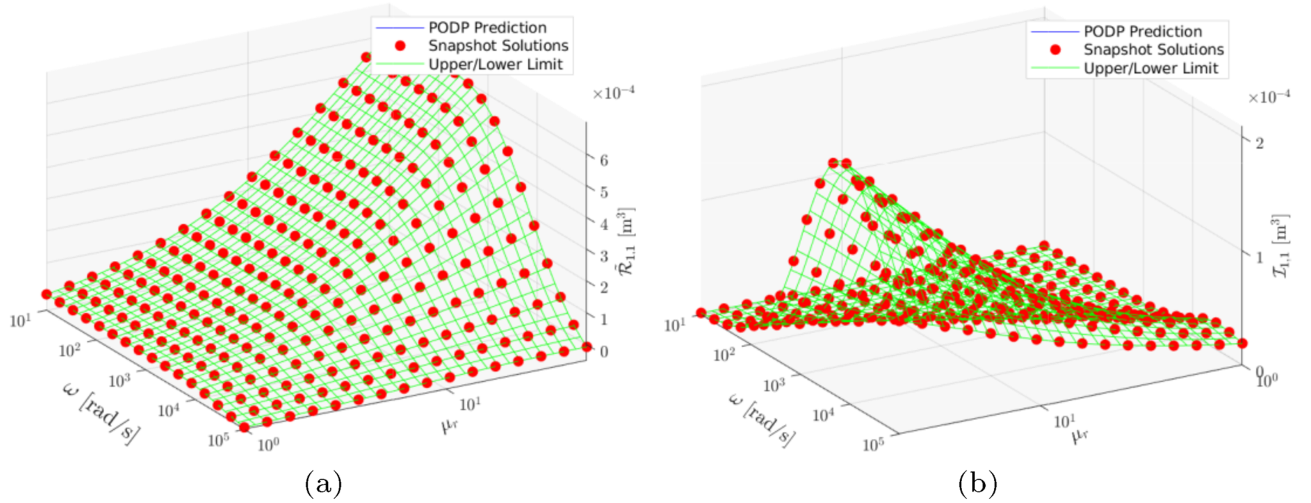


Fig. 14 Conducting permeable hammer head with $\alpha = 0.001$ m, $\sigma_* = 1.7 \times 10^6$ S/m, $1 \le \mu_r \le 50$ and $10^1 \le \omega \le 10^5$ rad/s: Results for (a) $(\tilde{\mathcal{R}}^{\text{PODP}} \pm \Delta)_{1,1}$, (b) $(\mathcal{I}^{\text{PODP}} \pm \Delta)_{1,1}$ using $N = 16^2$. On this scale, the upper and lower error certificates are indistinguishable from the PODP prediction

of interest, particularly in scrap metal sorting, if the full object description allowed by considering changes in μ_r can be applied for the estimation of an object's material properties.

Acknowledgements The authors gratefully acknowledge the financial support received from EPSRC in the form of grant EP/V009028/1.

Data availability The datasets used to for this paper are publicly available at https://zenodo.org/record/8131580 (DOI:10.5291/zenodo.8131580). This includes both numerical data and python source code. The datasets were generated using the open-source MPTCalculator software, available at https://github.com/MPT-Calculator/MPT-Calculator (InitialRelease branch).

Statements and Declarations

Financial or non-financial interests We gratefully acknowledge full funding received from the UK EPSRC under grant EP/V009028/1. The authors have no relevant financial or non-financial interests to disclose.

References

1. Ledger PD, Lionheart WRB (2020) The spectral properties of the magnetic polarizability tensor for metallic object characterisation. Math Methods Appl Sci 43(1):78–113. https://doi.org/10.1002/mma.5830
2. Ledger PD, Lionheart WRB (2015) Characterizing the shape and material properties of hidden targets from magnetic induction data. IMA J Appl Math 80(6):1776–1798. https://doi.org/10.1093/imamat/hxv015
3. Makkonen J, Marsh LA, Vihonen J, Jarvi A, Armitage DW, Visa A, Peyton AJ (2015) Improving reliability for classification of metallic objects using a WTMD portal. Meas Sci Technol. https://doi.org/10.1088/0957-0233/26/10/105103
4. Ledger PD, Lionheart WRB (2018) An explicit formula for the magnetic polarizability tensor for object characterization. IEEE Trans Geosci Remote Sens 56(6):3520–3533. https://doi.org/10.1109/TGRS.2018.2801359
5. Wilson BA, Ledger PD (2021) Efficient computation of the magnetic polarizabiltiy tensor spectral signature using proper orthogonal decomposition. Int J Numer Meth Eng 122(8):1940–1963. https://doi.org/10.1002/nme.6606
6. Davidson JL, Abdel-Rehim OA, Hu P, Marsh LA, O'Toole MD, Peyton AJ (2018) On the magnetic polarizability tensor of US coinage. Meas Sci Technol. https://doi.org/10.1088/1361-6501/aa9cf2
7. Özdeger T, Davidson JL, Verre WV, Marsh LA, Lionheart WRB, Peyton AJ (2021) Measuring the magnetic polarizability tensor using an axial multi-coil geometry. IEEE Sens J 21(17):19322–19333. https://doi.org/10.1109/JSEN.2021.3088809
8. Abdel-Rehim OA, Davidson JL, Marsh LA, O'Toole MD, Peyton AJ (2016) Magnetic polarizability tensor spectroscopy for low metal anti-personnel mine surrogates. IEEE Sens J 16(10):3775–3783. https://doi.org/10.1109/JSEN.2016.2535346
9. Özdeger T (2022) Advances in Techniques for the Characterisation of Targets in Metal Detection and Ultrawide Band Electromagnetic Screening Applications. Phd, The University of Manchester.

10. Özdeger T, Ledger PD, Peyton AJ (2022) A study on the magnetic polarizability tensors of minimum metal anti-personnel landmines. In: 2022 IEEE International Instrumentation and Measurement Technology Conference, pp. 0–5. IEEE, Ottawa, Canada. https://doi.org/10.1109/I2MTC48687.2022.9806662.
11. Hesthaven JS, Rozza G, Stamm B (2015) Certified Reduced Basis Methods for Parametrized Partial Differential Equations, 1st edn. Springer, Cham. https://doi.org/10.1007/978-3-319-22470-1
12. Lehrenfeld C (2010) Hybrid Discontinuous Galerkin Methods for Solving Incompressible Flow Problems. Masters, Rheinisch-Westfälischen Technischen Hochschule. https://doi.org/10.25625/O4VBYH
13. Schöberl J (1997) NETGEN an advancing front 2D/3D-mesh generator based on abstract rules. Comput Vis Sci 1(1):41–52. https://doi.org/10.1007/s007910050004
14. Schöberl J (2014) C++ 11 implementation of finite elements in NGSolve. Technical Report 30, Institute for Analysis and Scientific Computing, Vienna University of Technology, Vienna, Austria
15. Zaglmayr S (2006) High Order Finite Element Methods for Electromagnetic Field Computation. Phd, Johannes Kepler University. http://www.numa.uni-linz.ac.at/Teaching/PhD/Finished/zaglmayr
16. Ledger PD, Wilson BA, Amad AAS, Lionheart WRB (2021) Identification of metallic objects using spectral magnetic polarizability tensor signatures: object characterisation and invariants. Int J Numer Meth Eng 112(15):3941–3984. https://doi.org/10.1002/nme.6688
17. Wilson BA, Ledger PD, Lionheart WRB (2022) Identification of metallic objects using spectral magnetic polarizability tensor signatures: object classification. Int J Numer Meth Eng 123(9):2076–2111. https://doi.org/10.1002/nme.6927
18. Miah S, Sooriyakanthan Y, Ledger PD, Gil AJ, Mallett M (2023) Reduced order modelling using neural networks for predictive modelling of 3D-magneto-mechanical problems with application to magnetic resonance imaging scanners. Engineering with Computers, Accepted.
19. Hesthaven JS, Ubbiali S (2018) 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
20. Fresca S, Manzoni A (2022) POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. Comput Methods Appl Mech Eng 388:11845. https://doi.org/10.1016/j.cma.2021.114181
21. Ammari H, Chen J, Chen Z, Garnier J, Volkov D (2014) Target detection and characterization from electromagnetic induction data. J de Mathématiques Pures et Appliquées 101(1):54–75. https://doi.org/10.1016/j.matpur.2013.05.002
22. Bishop CM (2006) Pattern Recognition and Machine Learning. Springer, New York. https://doi.org/10.1007/978-3-030-57077-4_11
23. Scikit-Learn: sklearn.neural_network.MLPRegressor. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html#sklearn.neural_network.MLPRegressor. Accessed 04 Sept 2022
24. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-Learn: machine learning in Python. J Mach Learn Res 12:2825–2830
25. Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. Math Program 45(1):503–528. https://doi.org/10.1007/BF01589116
26. Wait JR (1951) A conducting sphere in a time varying magnetic field. Geophysics 16(4):666–672
27. Nocedal J, Wright SJ (1999) Numerical Optimization. Springer, New York. https://doi.org/10.1007/978-1-4471-2224-1_2
28. Mitchell BS (2004) An Introduction to Materials Engineering and Science for Chemical and Materials Engineers. Wiley, Hoboken