Massive Database Generation for 2.5D Borehole Electromagnetic Measurements using Refined Isogeometric Analysis

Ali Hashemian\textsuperscript{a,\,*}, Daniel Garcia\textsuperscript{b,c}, Jon Ander Rivera\textsuperscript{d,a}, David Pardo\textsuperscript{d,a,e}

\textsuperscript{a}BCAM – Basque Center for Applied Mathematics, Bilbao, Basque Country, Spain
\textsuperscript{b}CIMNE – International Center for Numerical Methods in Engineering, Barcelona, Cataluña, Spain
\textsuperscript{c}IDAIA – Institute of Environmental Assessment and Water Research, Barcelona, Cataluña, Spain
\textsuperscript{d}University of the Basque Country UPV/EHU, Leioa, Basque Country, Spain
\textsuperscript{e}Ikerbasque – Basque Foundation for Sciences, Bilbao, Basque Country, Spain

Abstract

Borehole resistivity measurements are routinely inverted in real-time during geosteering operations. The inversion process can be efficiently performed with the help of advanced artificial intelligence algorithms such as deep learning. These methods require a massive dataset that relates multiple Earth models with the corresponding borehole resistivity measurements. In here, we propose to use an advanced numerical method —refined isogeometric analysis (rIGA)— to perform rapid and accurate 2.5D simulations and generate databases when considering arbitrary 2D Earth models. Numerical results show that we can generate a meaningful synthetic database composed of 100,000 Earth models with the corresponding measurements in 56 hours using a workstation equipped with two CPUs.

Keywords: Geosteering; borehole resistivity measurements; refined isogeometric analysis; 2.5D numerical simulation; deep learning inversion.

1. Introduction

Geosteering plays a crucial role in oil and gas engineering. To perform geosteering operations, companies often employ borehole instruments that record electromagnetic (EM) data in real-time while drilling (Liu, 2017a). Since the electrical resistivity is highly sensitive to salinity, EM measurements are used to distinguish between hydrocarbon- and water-saturated rocks (Liu, 2017b).

Inversion techniques estimate layer-by-layer EM properties from the measurements, allowing for the adjustment of the logging trajectory during geosteering operations. Thus, they enable to select an optimal well trajectory toward the target hydrocarbon-saturated rocks. There exist a plethora of inversion methods in the literature, including gradient based methods (Vogel, 2002; Tarantola, 2005), statistics based methods (Watzenig, 2007; Kaipio and Somersalo, 2005), and artificial intelligence algorithms such as deep learning (Liu, 2017a).
2007; Shen et al., 2020), and artificial intelligence based methods (Kim and Nakata, 2018; Liu and Grana, 2019; Yang and Ma, 2019; Li et al., 2020). In geosteering EM measurements, deep learning (DL) methods with advanced encoder-decoder neural networks have recently demonstrated to be suitable to solve inverse problems (Shahriari et al., 2020a,b).

DL methods are fast, but require a massive training dataset. To decrease the online computational time during field operations, we often produce such a large dataset a priori (offline) using tens of thousands of simulations of borehole resistivity measurements (see, e.g., Kushnir et al., 2018). To generate the database for DL inversion, we employ simulation methods to solve Maxwell’s equations with different conductivity distributions (Earth models). Since 3D simulations are expensive and possibly unaffordable when computing such large databases, it is common to reduce the Earth model dimensionality to two or one spatial dimensions using a Fourier or a Hankel transform. These transformations lead to the so-called 2.5D (Abubakar et al., 2006; Pardo et al., 2008; Shen and Sun, 2008; Nam et al., 2013; Gernez et al., 2020) and 1.5D (Pardo and Torres-Verdún, 2015; Bakr et al., 2017; Shahriari and Pardo, 2020) formulations, respectively. 1.5D simulations are inaccurate when dealing with geological faults. In this work, we focus on the efficient generation of a massive database using 2.5D simulations —as a preliminary stage for deep learning inversions.

Galerkin methods are effective for simulating well-logging problems (see, e.g., Pardo et al., 2006, 2021; Calo et al., 2011; Ma et al., 2012; Wang et al., 2013; Rodríguez-Rozas et al., 2018; Chaumont-Frelet et al., 2018b; Nunes and Régis, 2020). Isogeometric analysis (IGA), introduced by Hughes et al. (2005), is a widely used Galerkin method for solving partial differential equations. IGA has been successfully employed in various electromagnetic (Buffa et al., 2010, 2014; Nguyen et al., 2012; Simpson et al., 2018; Simona et al., 2020) and geotechnical (Shahrokhabadi et al., 2019; Hageman et al., 2019) applications. IGA uses spline basis functions introduced in computer-aided design (CAD) as shape functions of finite element analysis (FEA). These basis functions exhibit high continuity (up to $C^{p-1}$, being $p$ the polynomial order of spline bases) across the element interfaces.

When comparing IGA and FEA, the former provides smoother solutions for wave propagation problems with a lower number of unknowns (Hughes et al., 2005; Cottrell et al., 2009). However, in contrast to the minimal interconnection of elements in FEA, high-continuity IGA discretizations strengthen the interconnection between elements, leading to an increase of the cost of matrix LU factorization per degree of freedom when using sparse direct solvers (Collier et al., 2012). In order to avoid this degradation and also benefit from the recursive partitioning capability of multifrontal direct solvers, Garcia et al. (2017) developed a new method called refined isogeometric analysis (rIGA). This discretization technique conserves desirable properties of high-continuity IGA discretizations, while it partitions the computational domain into blocks of macroelements weakly interconnected by low-continuity separators. As a result, the computational cost required for performing LU factorization decreases. The applicability of the rIGA framework to general EM problems was studied by Garcia et al. (2019). Compared to high-continuity IGA, rIGA produces solutions of EM problems up to $O(p^2)$ faster on large domains and close to $O(p)$ faster on small domains. rIGA also improves the approximation errors with respect to IGA since the continuity reduction of basis
functions increases the number of degrees of freedom (c.f. Cottrell et al., 2009) and enriches the Galerkin space.

Herein, we propose the use of rIGA discretizations to generate databases for DL inversion of 2.5D geosteering EM measurements. We consider a priori grids following the idea of optimal grid generation for 2.5D EM measurements presented by Rodríguez-Rozas et al. (2018) and the methods described by Rodríguez-Rozas and Pardo (2016) for Fourier mode selections. Compared to the FEA approach described by Rodríguez-Rozas et al. (2018) that assigns increasing polynomial orders for the elements near the well, we consider a (smooth) high-continuity IGA discretization with a fixed polynomial order everywhere and reduce the computational cost by continuity reduction of certain basis functions in the sense of rIGA framework. To assess the accuracy and computational efficiency of the rIGA approach in borehole resistivity simulations, we consider several model problems where high-angle wells cross spatially heterogeneous media exhibiting multiple geological faults. Then, we investigate the performance of the proposed approach when generating a synthetic database composed of 100,000 Earth models.

The remainder of this article is organized as follows. In Section 2, we review the governing equations of the 3D EM wave propagation problem and derive the 2.5D variational formulation. Section 3 introduces the considered borehole resistivity problem. In Section 4, we describe both high-continuity and refined isogeometric discretizations, followed by the implementation details in Section 5. We analyze the accuracy and computational efficiency of the rIGA approach when applied to borehole resistivity problems in Section 6. In this section, we also generate a database for DL inversion of geosteering measurements. Section 7 presents some discussions about the results and Section 8 draws the main conclusions and possible future research lines stemming from this work.

2. 2.5D variational formulation of EM measurements

2.1. 3D wave propagation problem

The two time-harmonic curl Maxwell’s equations describing the 3D wave propagation in an isotropic medium are

\[ \nabla \times \mathbf{E} + i\omega \mu \mathbf{H} = -i\omega \mu \mathbf{M}, \]
\[ \nabla \times \mathbf{H} = (\sigma + i\omega \varepsilon)\mathbf{E}, \]

where \( \mathbf{E} \) is the electric field, \( \mathbf{H} \) is the magnetic field, \( i \) is the imaginary unit, \( \sigma \) is the electric conductivity, \( \varepsilon \) is the electric permittivity, \( \mu \) is the magnetic permeability, \( \omega = 2\pi f \) is the angular frequency, with \( f \) being the transmitter frequency, and \( \mathbf{M} \) is the time-harmonic magnetic source located at \((x_0, y_0, z_0)\) and given by

\[ \mathbf{M} = \delta(x - x_0)\delta(y - y_0)\delta(z - z_0)[M_x, M_y, M_z] \quad \text{in } \mathbb{R}^3, \]

with \( \delta(\cdot) \) being the Dirac delta function defined as follows:

\[ \delta(x - x_0) = \begin{cases} \infty, & x = x_0, \\ 0, & x \neq x_0. \end{cases} \]
To incorporate an integrable approximation of the Dirac delta function, we consider a bell-like representation for the delta function. For example, in the $x$ direction, we approximate:

$$\delta(x - x_0) \approx \frac{1}{\alpha \sqrt{\pi}} \exp \left[ -\left( \frac{x - x_0}{\alpha} \right)^2 \right],$$

(5)

where $\alpha$ is a positive value.

From Maxwell’s equations, we obtain the following reduced wave formulation in terms of magnetic field $H$:

$$\begin{align*}
&\text{Find } H = [H_x, H_y, H_z], \text{ such that:} \\
&\nabla \times \left( \frac{1}{\sigma + i\omega \varepsilon} \nabla \times H \right) + i\omega \mu H = -i\omega \mu M \quad \text{in } \Omega, \\
&H \times n = 0 \quad \text{on } \partial \Omega,
\end{align*}$$

(6)

where $\Omega$ is the domain of study and $n$ is the unit normal (outward) vector on the boundary $\partial \Omega$. We define $\Omega$ as a tensor-product box:

$$\Omega = \Omega_x \times \Omega_y \times \Omega_z = (-L_x/2, L_x/2) \times (-L_y/2, L_y/2) \times (-L_z/2, L_z/2),$$

(7)

being $L_x, L_y,$ and $L_z$ positive real constants.

To introduce the weak formulation of this problem, we first define the $\mathbf{H}(\text{curl}; \Omega)$-conforming functional spaces

$$\mathbf{H}(\text{curl}; \Omega) := \left\{ W = [W_x, W_y, W_z] \in (L^2(\Omega))^3 : \nabla \times W \in (L^2(\Omega))^3 \right\},$$

(8)

$$\mathbf{H}_0(\text{curl}; \Omega) := \left\{ W \in \mathbf{H}(\text{curl}; \Omega) : W \times n = 0 \text{ on } \partial \Omega \right\}.$$  

(9)

The $\mathbf{H}(\text{curl}; \Omega)$ space is endowed with the inner product

$$(W, H)_{\mathbf{H}(\text{curl}; \Omega)} := (\nabla \times W, \nabla \times H)_{(L^2(\Omega))^3} + (W, H)_{(L^2(\Omega))^3},$$

$$:= \int_{\Omega} (\nabla \times W)^* : (\nabla \times H) \, d\Omega + \int_{\Omega} W^* : H \, d\Omega,$$

(10)

where $^*$ is the conjugate transpose of complex vector space and $:$ denotes the inner product.

We build the weak formulation by multiplying Eq. (6) with an arbitrary function $W \in \mathbf{H}_0(\text{curl}; \Omega)$, using Green’s formula, and integrating over the domain $\Omega$. The weak formulation is then

$$\begin{align*}
&\text{Find } H \in \mathbf{H}_0(\text{curl}; \Omega), \text{ such that for every } W \in \mathbf{H}_0(\text{curl}; \Omega), \\
&\nabla \times W, \frac{1}{\sigma + i\omega \varepsilon} \nabla \times H \bigg|_{(L^2(\Omega))^3} + i\omega \mu (W, H)_{(L^2(\Omega))^3} = -i\omega \mu (W, M)_{(L^2(\Omega))^3}.
\end{align*}$$

(11)

2.2. 2.5D variational formulation

Herein, we focus on the case when the material properties are homogeneous along one spatial direction, e.g., $y$-axis. We denote the domain for this case as $\Omega := \Omega_x \times \Omega_{y,z}$. We perform a Fourier transform along the $y$-axis to
represent the 3D problem as a sequence of uncoupled 2D problems, one per Fourier mode. In this case, we define the magnetic field $\mathbf{H}$ as a series expansion using the complex exponentials:

$$
\mathbf{H} := \sum_{\beta=-\infty}^{+\infty} \mathbf{H}_\beta \exp(i2\pi \beta y/L_y),
$$

where $\beta$ is the Fourier mode number and $\mathbf{H}_\beta = [H_{\beta x}, H_{\beta y}, H_{\beta z}]$ with $\mathbf{H}_\beta : \Omega_{s.z} \subset \mathbb{R}^2 \rightarrow \mathbb{C}^3$. Fourier modes satisfy the following orthogonality relationships, where $\delta_{ij}$ is the Kronecker delta:

$$
\frac{1}{L_y} \int_{-L_y/2}^{L_y/2} \exp(i2\pi \beta_1 y/L_y) \exp(i2\pi \beta_2 y/L_y) \, dy = \delta_{\beta_1 \beta_2}.
$$

By employing a test function of the form

$$
\mathbf{W} := \frac{1}{L_y} \mathbf{W}_\beta \exp(i2\pi \beta y/L_y),
$$

and defining the $\mathbf{H}(\text{curl}^\beta; \Omega_{s.z})$-conforming functional spaces

$$
\mathbf{H}(\text{curl}^\beta; \Omega_{s.z}) := \left\{ \mathbf{W}_\beta = [W_{\beta x}, W_{\beta y}, W_{\beta z}] \in (L^2(\Omega_{s.z}))^3 : W_{\beta y} \in H^1(\Omega_{s.z}) \text{ and } \nabla \times [W_{\beta x}, W_{\beta z}] \in (L^2(\Omega_{s.z}))^2 \right\},
$$

we build the following variational formulation from Eq. (6) by integrating over $\Omega_{s.z}$:

$$
\begin{align*}
\text{Find } \mathbf{H} &= \sum_{\beta=-\infty}^{+\infty} \mathbf{H}_\beta \exp(i2\pi \beta y/L_y), \quad \mathbf{H}_\beta \in \mathbf{H}_0(\text{curl}^\beta; \Omega_{s.z}) \\
\text{such that for every } \beta \in \mathbb{Z} \text{ and } \mathbf{W}_\beta \in \mathbf{H}_0(\text{curl}^\beta; \Omega_{s.z}),
\end{align*}
$$

$$
\left( \nabla \times \mathbf{W}_\beta, \frac{1}{\sigma + i\omega \varepsilon} \nabla \times \mathbf{H}_\beta \right)_{(L^2(\Omega_{s.z}))^3} + i\omega \mu \left( \mathbf{W}_\beta, \mathbf{H}_\beta \right)_{(L^2(\Omega_{s.z}))^3} = -i\omega \mu \left( \mathbf{W}_\beta, \mathbf{M}_\beta \right)_{(L^2(\Omega_{s.z}))^3},
$$

where

$$
\nabla \times \mathbf{W}_\beta := \left[ i\beta 2\pi \frac{dW_{\beta y}}{dz} - \frac{dW_{\beta x}}{dz}, \frac{dW_{\beta y}}{dx}, \frac{dW_{\beta z}}{dx} - i\beta \frac{2\pi}{L_y} W_{\beta z} \right].
$$

and $\mathbf{M}_\beta$ is the time-harmonic magnetic source written in terms of the Fourier transform as

$$
\mathbf{M}_\beta = \frac{1}{L_y} \delta(x - x_0) \delta(z - z_0) [M_x, M_y, M_z] \exp(i2\pi \beta y/L_y).
$$

This formulation corresponds to the 2.5D variational formulation previously described by, e.g., Rodríguez-Rozas et al. (2018) and Chaumont-Frelet et al. (2018b).

**Remark 1.** To solve the variational problem of Eq. (17), we require an appropriate space in $\Omega_{s.z}$ over which $\mathbf{W}_\beta$, $\nabla \times [W_{\beta x}, W_{\beta z}]$, and $\nabla W_{\beta y}$ are integrable, i.e., $[W_{\beta x}, W_{\beta z}] \in \mathbf{H}(\text{curl}; \Omega_{s.z})$ and $W_{\beta y} \in H^1(\Omega_{s.z})$. Thus, we use the $\mathbf{H}(\text{curl}^\beta; \Omega_{s.z})$ solution space —equivalent to the $\mathbf{H}(\text{curl}; \Omega_{s.z}) \times H^1(\Omega_{s.z})$ mixed space— that fulfills the mentioned requirements.
3. Borehole resistivity measurement acquisition system

We consider a logging-while-drilling (LWD) instrument equipped with transmitters \((T_i)\) and receivers \((R_j)\). This tool is sensitive to resistivities within the range \(0.2 \sim 500 \, \Omega \cdot \text{m} \) (phase resistivity) and \(0.2 \sim 300 \, \Omega \cdot \text{m} \) (amplitude resistivity) under an operating frequency between 0.1 and 2 MHz (Liu, 2017a). For the sake of simplicity, herein, we restrict to two transmitters and two receivers symmetrically located around the tool center (see Fig. 1) at an operating frequency of 2 MHz.

![Fig. 1. A schematic LWD instrument with two transmitters and two receivers symmetrically located around the tool center.](image)

Triaxial logging instruments generate measurements for all possible orientations of the transmitter–receiver pairs. We follow the notation presented by Davydycheva (2011) and Rodríguez-Rozas et al. (2018) to denote the magnetic field. Thus, we write \(H_{\text{ZZ}}^{T_i R_j} \in \mathbb{C}\) as the coaxial magnetic field in the borehole system of coordinates induced by transmitter \(T_i\) and measured at receiver \(R_j\) \((i, j = 1, 2)\). We use the magnetic fields measured at \(R_1\) and \(R_2\) to compute the attenuation ratio and phase difference. We symmetrize the signal originating from \(T_1\) and \(T_2\) to obtain the quantity of interest \(Q_{\text{ZZ}}\) at each logging position as

\[
Q_{\text{ZZ}} := \frac{1}{2} \left( \log \frac{H_{\text{ZZ}}^{T_1 R_1}}{H_{\text{ZZ}}^{T_2 R_2}} + \log \frac{H_{\text{ZZ}}^{T_2 R_1}}{H_{\text{ZZ}}^{T_1 R_2}} \right). \tag{20}
\]

Then, we compute the attenuation ratio \((A)\) and phase difference \((P)\), respectively, as the real and imaginary parts of \(Q_{\text{ZZ}}\):

\[
A := \text{Re}(Q_{\text{ZZ}}), \tag{21}
\]

\[
P := \text{Im}(Q_{\text{ZZ}}). \tag{22}
\]

We can then obtain the apparent resistivities based on the attenuation ratio and phase difference \((\rho_A\) and \(\rho_P\), respectively) using a look-up table algorithm. This algorithm obtains the apparent resistivities from the tool response in a homogeneous isotropic medium, which is analytically known (see Anderson, 2001).

4. Refined isogeometric analysis

In this work, we consider a multi-field EM problem and discretize the 2.5D variational formulation of Eq. (17) using a B-spline generalization of a curl-conforming space, introduced by Buffa et al. (2010). We first review some basic concepts of high-continuity IGA discretizations.
4.1. High-continuity IGA discretization

Given the parametric domain \([\xi, \zeta \in \tilde{\Omega}_{xz} : (0, 1)^2 \subset \mathbb{R}^2]\), we introduce the spline space \(S_{p,k}^{p_r} \) as

\[
S_{p,k}^{p_r} := \text{span} \left\{ B_{i,j}^{p,k} \right\}_{i=0,j=0}^{n_r-1,n_k-1},
\]

where \(n, p, \) and \(k\) with their indices are the number of degrees of freedom, polynomial degree, and continuity of basis functions in \(x\) and \(z\) directions, respectively, resulting in \(n := n - p\) number of elements in each direction. The bivariate basis functions are

\[
B_{i,j}^{p,k} := B_i^p(\xi) \otimes B_j^k(\zeta), \quad i = 0, \ldots, n_x - 1, \quad j = 0, \ldots, n_z - 1,
\]

where the univariate bases are expressed by the Cox–De Boor recursion formula (Piegl and Tiller, 1997) as

\[
B_i^p(\xi) = \begin{cases} 1, & \xi_i \leq \xi < \xi_{i+1}, \\ 0, & \text{otherwise}, \end{cases}
\]

\[
B_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_i^{p-1}(\xi) + \frac{\xi_{i+p} - \xi}{\xi_{i+p} - \xi_{i+1}} B_{i+1}^{p-1}(\xi),
\]

and spanned over the respective knot sequences in \(x\) and \(z\) directions, given by

\[
\Xi = [0, 0, \ldots, 0, \xi_{p+1}, \xi_{p+2}, \ldots, \xi_{n_x-1}, 1, 1, \ldots, 1],
\]

\[
\hat{\xi}_p = \{ \xi_{i+p} \}_{i=0}^{n_x-1},
\]

\[
Z = [0, 0, \ldots, 0, \zeta_{p+1}, \zeta_{p+2}, \ldots, \zeta_{n_z-1}, 1, 1, \ldots, 1],
\]

\[
\hat{\zeta}_p = \{ \zeta_{i+p} \}_{i=0}^{n_z-1}.
\]

We assume single multiplicities for all knots, providing maximum continuity \(k = p - 1\) for the IGA discretization.

Fig. 2 illustrates the \(H(\text{curl}) \times H^1\) IGA discrete space in \(\Omega_{xz}\) along with the univariate basis functions of the respective vector fields. For brevity, herein and in the following, we exclude the superscript \(\beta\) in referring to the components of the magnetic field.

We define the spaces in the parametric domain and introduce the appropriate transformations to obtain the discretization on the physical domain. We start with the set of discrete spaces in the parametric domain, given by

\[
\hat{Y}_{h}^{\text{curl}}(\tilde{\Omega}_{xz}) := S_{k-1}^{p-1} \times S_{k-1}^{p-1},
\]

\[
\hat{Q}_{h}^{\text{grad}}(\tilde{\Omega}_{xz}) := S_{k}^{p}.
\]

By defining \(F : \tilde{\Omega}_{xz} \rightarrow \Omega_{xz}\) as the geometric mapping from the parametric domain onto the physical domain, and \(DF\) as its Jacobian, we introduce the set of discrete spaces in the physical domain:

\[
Y_{h}^{\text{curl}}(\Omega_{xz}) := \left\{ \mathbf{H}_{xz} = [H_x, H_z] \in H(\text{curl}; \Omega_{xz}) \cap H^1(\Omega_{xz}) : t^{\text{curl}}(\mathbf{H}_{xz}) = \mathbf{H}_{xz} \in \hat{Y}_{h}^{\text{curl}}(\tilde{\Omega}_{xz}) \right\},
\]

\[
Q_{h}^{\text{grad}}(\Omega_{xz}) := \left\{ H_y \in H^1(\Omega_{xz}) : \mathbf{t}^{\text{grad}}(H_y) = \nabla H_y \in \hat{Q}_{h}^{\text{grad}}(\tilde{\Omega}_{xz}) \right\},
\]

7
where we use the following curl- and grad-preserving pullback mappings (Garcia et al., 2019; Buffa et al., 2010):

\[ \iota^{\text{curl}}(H_{z,c}) := (DF)^T(H_{z,c} \circ F), \]
\[ \iota^{\text{grad}}(H_y) := H_y \circ F. \]

Thus, by defining the discrete space
\[ H_{h,0}(\Omega_{x,z}) := \{ H_{\beta,h} \in V_{h}^{\text{curl}}(\Omega_{x,z}) \times Q_{h}^{\text{grad}}(\Omega_{x,z}) : H_{\beta,h} \times n = 0 \text{ on } \partial \Omega \}, \]
we write the discrete form of Eq. (17) as follows (subscript \( h \) refers to discrete solution):

\[
\begin{cases}
\text{Find } H_h = \sum_{\beta=-\infty}^{+\infty} H_{\beta,h} \exp(i2\pi \beta y/L_y), & H_{\beta,h} \in H_{h,0}(\Omega_{x,z}) \\
\text{such that for every } \beta \in \mathbb{Z} \text{ and } W_{\beta,h} \in H_{h,0}(\Omega_{x,z}), \\
\left( \nabla \times W_{\beta,h}, \frac{1}{\sigma + i\omega \varepsilon} \nabla \times H_{\beta,h} \right)_{L^2(\Omega_{x,z})^3} + i\omega \mu \left( W_{\beta,h}, H_{\beta,h} \right)_{L^2(\Omega_{x,z})^3} = -i\omega \mu \left( W_{\beta,h}, M_{\beta} \right)_{L^2(\Omega_{x,z})^3}.
\end{cases}
\]
number of elements in the domain. Garcia et al. (2017) first presented this strategy for $H^1$ spaces and then extended it to $H$(curl), $H$(div), and $L^2$ spaces (see Garcia et al., 2019). Starting from the high-continuity $C^{p-1}$ IGA discretization, rIGA reduces the continuity of certain basis functions by increasing the multiplicity of the respective existing knots. Hence, the computational domain is subdivided into high-continuity macroelements interconnected by low-continuity hyperplanes. These hyperplanes coincide with the locations of the separators at different partitioning levels of the multifrontal direct solvers. Thus, rIGA reduces the computational cost of matrix factorization when solving PDE systems in comparison to IGA and FEA.

For multi-field problems discretized using $H$(curl; $\Omega \times z$) $\times H^1$( $\Omega \times z$) spaces, we preserve the commutativity of the de Rham diagram (Demkowicz et al., 2000) by reducing the continuity in $k-1$ degrees. To achieve this, we use both $C^0$ and $C^1$ hyperplanes and reduce the continuity across the interface between the subdomains (i.e., macroelements). Fig. 3 depicts the rIGA discretization of the $H$(curl; $\Omega \times z$) $\times H^1$( $\Omega \times z$) space of Fig. 2 after one level of symmetric partitioning, which results in macroelements containing $4 \times 4$ elements.

Previous works show rIGA discretizations provide significant improvements in the solution time and memory requirements. In particular, the rIGA solution is obtained up to $O(p^3)$ faster in large domains — and $O(p)$ faster in small domains — than the IGA solution. In comparison to traditional FEA with the same number of elements, rIGA
provides even larger improvements. rIGA also reduces the memory requirements since the rIGA LU factors have fewer nonzero entries than the IGA LU factors. Finally, rIGA improves the approximation error with respect to IGA since the continuity reduction of basis functions enriches the Galerkin space (see Garcia et al., 2017, 2019; Hashemian et al., 2021).

5. Implementation details

We implement discrete $H(\text{curl}; \Omega_x, z) \times H^1(\Omega_x, z)$ spaces using PetIGA-MF (Sarmiento et al., 2017), a multi-field extension of PetIGA (Dalcin et al., 2016), a high-performance isogeometric analysis implementation based on PETSc (portable extensible toolkit for scientific computation) (Balay et al., 1997). PetIGA-MF allows the use of different spaces for each field of interest and employs data management libraries to condense the data of multiple fields in a single object, thus simplifying the discretization construction. This framework also allows us to investigate both IGA and rIGA discretizations in our 2.5D problem with different numbers of elements, different polynomial degrees of the B-spline spaces, and different partitioning levels of the mesh.

We use Intel MKL with PARDISO (Petra et al., 2014a,b) as our sparse direct solver package to construct LU factors for solving the linear systems of equations. PARDISO employs supernode techniques to perform the matrix factorization (see, e.g., Schenk et al., 2000; Schenk and Gärtner, 2004). It provides parallel factorization using OpenMP directives (Dagum and Menon, 1998) and uses the automatic matrix reordering provided by METIS (Karypis and Kumar, 1998). We executed all tests on a workstation equipped with two Intel Xeon Gold 6230 CPUs at 2.10 GHz with 40 threads per CPU.

We employ a tensor-product mesh with variable element sizes (see Fig. 4). At each logging position, the computational mesh has a fine subgrid in the central part of the domain with element size equal to $h \times h$. This subgrid is surrounded by another tensor-product grid whose element sizes grow slowly until reaching the boundary. Let $n_e$ be the number of elements in each direction and $n_c < n_e$ is the number of elements in each direction located at the central part of the domain. We use the power function of Eq. (37) to model the geometrical progression of the mesh. We follow the algorithm presented by Rodríguez-Rozas et al. (2018) to find suitable values for $n_c$ and growth rate $r$. Starting from the center point of a symmetric domain, we obtain the size of the $i$-th element in each direction $h_i$ ($i = 1, 2, ..., n_e/2$) as follows:

$$h_i = \begin{cases} 
  h, & 1 \leq i \leq n_c/2, \\
  h r^{(i-n_c/2)}, & n_c/2 < i \leq n_e/2.
\end{cases}$$ (37)

Remark 2. For each logging position, we perform a single symbolic factorization common to all Fourier modes, followed by a numerical factorization per Fourier mode. Once we solve the system of equations for the first transmitter, we update the right-hand side of Eq. (36) to solve for the magnetic field induced by the second transmitter and use backward substitution. Hence, we perform only one LU factorization per Fourier mode per logging position for both transmitters.
Remark 3. The convergence of the Fourier series leads to a fast decay of the real and imaginary parts of $H_{ZZ}$ for higher Fourier modes (see Rodríguez-Rozas and Pardo, 2016). Thus, we truncate the series of Eq. (12) when the magnetic field at the receivers is sufficiently small, such that $\beta \in [-N_f, N_f]$, being $2N_f + 1$ the total number of Fourier modes. Due to the symmetry of the media along the $y$ direction, we only consider $\beta \in [0, N_f]$.

6. Numerical Results

In this section, we first assess the accuracy of the rIGA approach in a homogeneous medium. We also investigate the computational efficiency of the rIGA framework in comparison with IGA and FEA approaches. Then, we consider two model problems consisting of high-angle wells crossing spatially heterogeneous media with multiple geological faults. Finally, we produce our synthetic training dataset as a preliminary stage for DL inversion (see Appendix A for more details regarding the numerical codes and required packages). In our simulations, we consider one operational mode of a commercial logging tool (Zhou, 2016) with $l_R = 10.16$ cm and $l_T = 56.8325$ cm (see Fig. 1). We select the free space electric permittivity and magnetic permeability as $\varepsilon = 8.85 \times 10^{-12} \text{ F} \cdot \text{m}^{-1}$ and $\mu = 4\pi \times 10^{-7} \text{ N} \cdot \text{A}^{-2}$, respectively. We also consider a transmitter frequency $f = 2$ MHz.

6.1. Homogeneous medium

We assume the logging instrument is placed in a homogeneous medium with resistivity $\rho = 1/\sigma = 100 \Omega \cdot \text{m}$. This high-resistivity case is numerically more challenging than low-resistivity cases since it requires a larger number of Fourier modes and numerical precision. We consider a cube domain of length $L = 18$ m for our first case study.
6.1.1. Accuracy assessment

To assess the accuracy and select certain discretization parameters, we compare the numerical attenuation ratio, \(A\), and phase difference, \(P\), given by Eqs. (21) and (22), with the expected (i.e., exact) values, \(A_e\) and \(P_e\), obtained from \(\rho_e = 1/\sigma\). Fig. 5 shows the numerical errors, i.e., \(|1 - A/A_e|\) and \(|1 - P/P_e|\), as a function of the number of Fourier modes when computing attenuation ratio and phase difference in a homogeneous medium. Herein, we select a domain with \(64 \times 64\) elements to ensure a fast numerical solution for our measurements. We compare the results of the high-continuity \(C^{p-1}\) IGA with FEA and also with an rIGA discretization that employs \(8 \times 8\) macroelements. This macroelement size provides the fastest results for moderate size domains (see Garcia et al., 2017). We consider three different mesh sizes — \(h = 0.025\) m, 0.033 m, and 0.050 m — and different polynomial degrees — \(p = 3, 4,\) and 5. The best results correspond to \(h = 0.025\) m (blue lines in the figure). We also observe that rIGA and FEA discretizations deliver lower errors compared to their IGA counterparts — when the same number of elements and polynomial degree are considered — taking into account that rIGA provides solutions with higher computational efficiency (see Section 6.1.2).

Fig. 5. Numerical errors when computing attenuation ratio, \(A\), and phase difference, \(P\), in a homogeneous medium using IGA and rIGA discretizations, obtained by for a \(64 \times 64\) mesh with different element sizes \(h\) and polynomial degrees \(p\).

To investigate the decay of the solution for each Fourier mode, we compare numerical results with the analytical 2.5D solution in the homogeneous medium presented by Rodríguez-Rozas et al. (2018). In particular, given \(M_z\) as the only nonzero component of the magnetic source, it is possible to analytically determine the coaxial magnetic field for each Fourier mode as follows:
\[ H_{ZZ}(\beta) = -i\omega \mu \sigma \tau_{\beta_i} + \frac{\partial^2 \tau_{\beta_i}}{\partial \tau^2}, \]  

(38)

with

\[ \tau_{\beta_i} = \frac{M_z}{2\pi} \frac{1}{L_y} K_0(CR) \exp(i2\pi y_0/L_y), \]  

(39)

where \( K_0(\cdot) \) is the modified Bessel function of the second kind of order zero, and

\[ C = \left( \frac{2\pi \beta}{L_y} \right)^2 + i\omega \mu \sigma, \]  

(40)

\[ R = \sqrt{(x-x_0)^2 + (z-z_0)^2}. \]  

(41)

Fig. 6 compares the decay of the numerical coaxial magnetic field \( H_{ZZ}(\beta) \) with its analytical counterpart for some Fourier modes. Using a domain with \( 64 \times 64 \) elements and \( h = 0.025 \text{ m} \), we monitor the decay of the propagated waves at distances within the interval \([0.2, 1.0]\) m from the transmitters to ensure that the solutions at both receivers properly approximate the analytical ones. Results show that rIGA discretizations deliver increased accuracy for all tested polynomial degrees.

![Fig. 6](image-url)

**Fig. 6.** Comparison of the decay of the numerical and analytical coaxial magnetic fields for some Fourier modes, obtained in a grid of \( 64 \times 64 \) elements with \( h = 0.025 \text{ m} \) and different polynomial degrees.

6.1.2. **Computational efficiency**

García et al. (2017, 2019) provide theoretical cost estimates of solving \( H^1 \) and \( H(\text{curl}) \) discrete spaces, respectively. Herein, we add these estimates to predict the cost of discretizing the \( H(\text{curl}; \Omega_{x,z}) \times H^1(\Omega_{x,z}) \) space appearing in our 2.5D EM problem. We conclude that the cost of LU factorization of the rIGA matrix for this combined space is between \( O(p) \) and \( O(p^2) \) times smaller than that for IGA. Details are omitted for the sake of simplicity.

To numerically assess the computational efficiency confirming the aforementioned theoretical results, we consider two different grids in \( \Omega_{x,z} \) with \( 64 \times 64 \) and \( 128 \times 128 \) elements, respectively. Using continuity reduction, we split
the mesh symmetrically into macroelements whose sizes are powers of two. In this context, the maximum-continuity \( C^{p-1} \) IGA discretization is composed of one macroelement containing the entire grid, while \( C^0 \) FEA with minimum continuity across all element interfaces is composed of macroelements that contain only one element. Fig. 7 shows the number of FLOPs and time required to solve the borehole resistivity problem for each Fourier mode per logging position. We compare the computational costs for different polynomial degrees and different continuity reduction levels of basis functions. The cost of rIGA reaches the minimum with \( 8 \times 8 \) macroelements almost in all cases, confirming the theoretical estimates obtained from the results of Garcia et al. (2017).

Fig. 7. Computational cost in terms of FLOPs and time for solving a 2.5D borehole resistivity problem per logging position per Fourier mode. We test rIGA discretizations with two different grids of \( 64 \times 64 \) and \( 128 \times 128 \) elements. The computational times correspond to the use of parallel solver PARDISO using two threads.
Our numerical tests show that for a moderate size 2.5D problem, the reduction in the number of FLOPs is $O(p)$ with respect to IGA. When compared to FEA, rIGA delivers larger improvement factors. These improvement factors in terms of FLOPs also hold in terms of time when performing a sequential factorization. In our parallel PARDISO solver, we observe a small degradation of the rIGA improvement factors in terms of times in comparison to those obtained in terms of FLOPs (see Table 1).

Table 1. Computational cost for the 2.5D borehole resistivity measurements per logging position per Fourier mode. We report the solution time and FLOPs when using $C^{p-1}$ IGA, rIGA with $8 \times 8$ macroelements, and $C^0$ FEA with the same number of elements and polynomial degree. The computational times correspond to the use of parallel solver PARDISO using two threads.

<table>
<thead>
<tr>
<th>Domain size</th>
<th>Polynomial degree</th>
<th>Discretization method</th>
<th>Number of FLOPs</th>
<th>Improvement factor (FLOPs)</th>
<th>Time (s)</th>
<th>Improvement factor (time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>IGA</td>
<td>6.56e+10</td>
<td></td>
<td></td>
<td>0.407</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rIGA</td>
<td>2.85e+10</td>
<td>IGA/rIGA</td>
<td>2.30</td>
<td>0.242</td>
<td>IGA/rIGA 1.68</td>
</tr>
<tr>
<td></td>
<td>FEA</td>
<td>1.65e+11</td>
<td>FEA/rIGA</td>
<td>5.79</td>
<td>0.999</td>
<td>FEA/rIGA 4.12</td>
</tr>
<tr>
<td>64×64</td>
<td>IGA</td>
<td>1.62e+11</td>
<td></td>
<td></td>
<td>0.875</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rIGA</td>
<td>4.97e+10</td>
<td>IGA/rIGA</td>
<td>3.26</td>
<td>0.419</td>
<td>IGA/rIGA 2.09</td>
</tr>
<tr>
<td></td>
<td>FEA</td>
<td>5.56e+11</td>
<td>FEA/rIGA</td>
<td>11.19</td>
<td>3.061</td>
<td>FEA/rIGA 7.29</td>
</tr>
<tr>
<td></td>
<td>IGA</td>
<td>3.10e+11</td>
<td></td>
<td></td>
<td>1.645</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>rIGA</td>
<td>7.33e+10</td>
<td>IGA/rIGA</td>
<td>4.23</td>
<td>0.620</td>
<td>IGA/rIGA 2.65</td>
</tr>
<tr>
<td></td>
<td>FEA</td>
<td>1.37e+12</td>
<td>FEA/rIGA</td>
<td>18.69</td>
<td>6.806</td>
<td>FEA/rIGA 10.98</td>
</tr>
<tr>
<td>3</td>
<td>IGA</td>
<td>5.72e+11</td>
<td></td>
<td></td>
<td>3.144</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rIGA</td>
<td>2.24e+11</td>
<td>IGA/rIGA</td>
<td>2.55</td>
<td>1.423</td>
<td>IGA/rIGA 2.21</td>
</tr>
<tr>
<td></td>
<td>FEA</td>
<td>1.31e+12</td>
<td>FEA/rIGA</td>
<td>5.85</td>
<td>7.456</td>
<td>FEA/rIGA 5.24</td>
</tr>
<tr>
<td>128×128</td>
<td>IGA</td>
<td>1.43e+12</td>
<td></td>
<td></td>
<td>6.903</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rIGA</td>
<td>3.56e+11</td>
<td>IGA/rIGA</td>
<td>4.02</td>
<td>2.495</td>
<td>IGA/rIGA 2.77</td>
</tr>
<tr>
<td></td>
<td>FEA</td>
<td>4.44e+12</td>
<td>FEA/rIGA</td>
<td>12.47</td>
<td>22.885</td>
<td>FEA/rIGA 9.17</td>
</tr>
<tr>
<td></td>
<td>IGA</td>
<td>2.82e+12</td>
<td></td>
<td></td>
<td>12.911</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>rIGA</td>
<td>4.97e+11</td>
<td>IGA/rIGA</td>
<td>5.67</td>
<td>3.305</td>
<td>IGA/rIGA 3.91</td>
</tr>
<tr>
<td></td>
<td>FEA</td>
<td>–</td>
<td>FEA/rIGA</td>
<td>–</td>
<td>–</td>
<td>FEA/rIGA –</td>
</tr>
</tbody>
</table>

6.2. Heterogeneous media

We further examine the accuracy of our rIGA approximation over two synthetic heterogeneous model problems.

6.2.1. One geological fault

We consider the model problem of Fig. 8 with a constant dip angle of 80°. We consider the LWD instrument described in Section 3 and simulate measurements recorded over 200 equally-spaced logging positions throughout the well trajectory.
Fig. 8. Model problem with a constant dip angle of 80° passing through a geological fault and three different materials (well trajectory is highlighted by a red dashed line). Dimensions are in meters.

Fig. 9 shows the apparent resistivities based on the attenuation ratio and phase difference ($\rho_A$ and $\rho_P$, respectively). We obtain the results using $N_f = 70$ and an rIGA discretization with $64 \times 64$ elements, $p = 4$, and $8 \times 8$ macroelements. Results are in good agreement with those presented by Rodríguez-Rozas et al. (2018).

6.2.2. Two geological faults and inclined layers

Fig. 10 shows the second model problem containing two geological faults and inclined layers. The logging trajectory starts from a sandstone layer with a resistivity of $\rho = 3 \ \Omega \cdot m$, and passes through an oil-saturated layer with $\rho = 100 \ \Omega \cdot m$. The tool trajectory also passes through a water-saturated layer with $\rho = 0.5 \ \Omega \cdot m$.

In particular, inclined layers produce the so-called staircase approximations (Cangellaris and Wright, 1991). This phenomenon occurs because the physical interfaces of the conductivity model are not aligned with the element edges. Thus, the conductivity parameter takes different values inside some elements of the mesh. To tackle this issue, discretization techniques using nonfitting grids (Chaumont-Frelet et al., 2018a,b) are available, but they have not been considered here for simplicity.

Fig. 11 shows the apparent resistivities based on the attenuation ratio and phase difference throughout the logging trajectory.
trajectory and compares their value with the exact resistivity. We simulate the resistivities at 1,080 logging positions with \( N_f = 70 \). We use an rIGA discretization with 64 × 64 elements, \( p = 4 \), and 8 × 8 macroelements.

6.3. Database generation for DL inversion

To produce our synthetic training dataset for DL inversion, we consider heterogeneous medium containing three different layers and six varying parameters at each logging position, as described in Fig. 12 and Table 2. We select three different electrical conductivities: \( \sigma_c \) for the central layer, and \( \sigma_u \) and \( \sigma_l \) for the upper and lower layers, respectively. We assume the tool center is always within the middle layer and has vertical distances of \( d_u \) and \( d_l \) from the upper and lower layers, respectively. The sixth varying parameter is the dip angle, \( \varphi \), measured from the vertical direction.

In here, we create a dataset of 100,000 samples and compute the apparent resistivities obtained from random com-
Table 2. Varying parameters employed to generate the training dataset for DL inversion.

<table>
<thead>
<tr>
<th>Varying parameters</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical conductivity of the central layer</td>
<td>$\log_{10}(\sigma_c)$</td>
</tr>
<tr>
<td>Electrical conductivity of the upper layer</td>
<td>$\log_{10}(\sigma_u)$</td>
</tr>
<tr>
<td>Electrical conductivity of the lower layer</td>
<td>$\log_{10}(\sigma_l)$</td>
</tr>
<tr>
<td>Distance of the tool center from the upper layer</td>
<td>$\log_{10}(d_u)$</td>
</tr>
<tr>
<td>Distance of the tool center from the lower layer</td>
<td>$\log_{10}(d_l)$</td>
</tr>
<tr>
<td>Dip angle between the tool and the layered media</td>
<td>$\varphi$</td>
</tr>
</tbody>
</table>

Varying parameters at each logging position when producing the training dataset for DL inversion.

Fig. 12. Varying parameters at each logging position when producing the training dataset for DL inversion.

Table 2. Varying parameters employed to generate the training dataset for DL inversion.

<table>
<thead>
<tr>
<th>Varying parameters</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical conductivity of the central layer</td>
<td>$\log_{10}(\sigma_c)$</td>
</tr>
<tr>
<td>Electrical conductivity of the upper layer</td>
<td>$\log_{10}(\sigma_u)$</td>
</tr>
<tr>
<td>Electrical conductivity of the lower layer</td>
<td>$\log_{10}(\sigma_l)$</td>
</tr>
<tr>
<td>Distance of the tool center from the upper layer</td>
<td>$\log_{10}(d_u)$</td>
</tr>
<tr>
<td>Distance of the tool center from the lower layer</td>
<td>$\log_{10}(d_l)$</td>
</tr>
<tr>
<td>Dip angle between the tool and the layered media</td>
<td>$\varphi$</td>
</tr>
</tbody>
</table>

Varying parameters within a given range of resistivities $\rho = 1/\sigma \in [1, 100]$ $\Omega \cdot \text{m}$ (see Table 2). For generating the dataset, we use two different types of parallelization. One parallelization is related to the parallel factorization of the direct solver, and the other is the trivial parallelization based on scheduling the solutions of independent Earth models onto different processors. Using 40 threads, we solve for 20 different Earth models, each executing over two threads. Table 1 shows that the required time for matrix factorization of the 2.5D EM problem using optimal rIGA discretization with $64 \times 64$ grid, $p = 4$, and $8 \times 8$ macroelements is about 0.42 seconds per Fourier mode. Considering $N_f = 70$, and the additional time required for pre/postprocessing and inter-thread communications, each set of independent runs (consists of 20 different Earth models) takes about 40 seconds. Thus, we perform 5,000 sequential runs to construct our 100,000 samples in about 56 hours. To create a larger database, we could execute over a cluster of hundreds of CPUs/threads, expecting a perfect parallel scalability.

Fig. 13a depicts the graphs of attenuation ratio, $A$, versus phase difference, $P$, obtained from the 100,000 Earth models when using rIGA discretization for generating the database. Since there is a strong correlation between $A$ and $P$, the data distribution on the plot follows an almost straight line. We also display in Fig. 13b the correlation between apparent resistivities based on attenuation ratio and phase difference.
Phase difference, \( P \).

Attenuation ratio, \( A \).

Apparent resistivity based on phase, \( \rho_P (\Omega \cdot m) \).

Apparent resistivity based on attenuation, \( \rho_A (\Omega \cdot m) \).

Fig. 13. (a) Attenuation ratio vs. phase difference, and (b) apparent resistivity based on attenuation vs. apparent resistivity based on phase, obtained for the 100,000 Earth models. We use rIGA discretization with 64 × 64 elements, \( p = 4 \), and 8 × 8 macroelements for generating the database.

7. Discussions

Herein, we discuss some of the topics observed during the course of this research:

**FEA vs. IGA and rIGA.** When using FEA, the solution space is characterized by basis functions that have support over up to two elements in each spatial direction. Considering \( n_e \) as the number of elements in each direction, \( p \) as the polynomial degree of basis functions, and \( d \) as the space dimension, we deal with a linear system of equations that has \( O((n_e p)^d) \) unknowns. Whereas in IGA, each basis function is spanned over \( p + 1 \) elements. As a result, not only we have smoother solution space using IGA (because of higher continuity of basis functions), but also the system of equations has lower number of unknowns, namely, \( O((n_e + p)^d) \), resulting in cheaper computations compared to FEA. In practice, if we want to reduce the FEA computational cost, we may decrease either the degree \( p \) or the number of elements \( n_e \). Either of these deteriorates the solution accuracy. Herein, we propose to use the “refined IGA” (rIGA). Thus, we conserve the desirable properties of IGA while reducing the solution cost of direct solvers by decreasing the interconnection of elements in system matrices.

**Different types of parallelization.** One may consider different types of parallelization, e.g., along the Fourier modes, parallel factorization using parallel direct solvers, and the trivial parallelization based on independent Earth models. In this work, we consider the latter two: parallelizations related to the parallel factorization, and scheduling the solutions of independent Earth models onto different processors. Since each new Earth model entails a new symbolic factorization, the advantage of using these parallelizations lies in the fact that we can use one symbolic factorization for all Fourier modes associated with the same processor.
Synthetic vs. real dataset. We employ synthetic data because we lack from massive annotated real data, i.e., inverted Earth models. Real data is typically sparse (i.e., insufficient) to train a neural network for DL inversion. Nonetheless, we may complement the synthetic data with the real data to benefit from both. From the computational point of view, borehole measurements satisfy Maxwell’s equations. However, due to instrumentation noise and geometrical complexity, we may be unable to use real data in some occasions and, instead, we employ synthetic data to reproduce real data. It is usual in the industry to employ synthetic data for these purposes and also to add noise to the synthetic measurements. In principle, it is possible to use a transfer learning algorithm to adapt the synthetic data to the real data and then, compensate for the noise using a few real samples and a pretrained model based on the synthetic data.

2D Earth models for database generation. Using more detailed 2D Earth models do not affect the computational costs, but may affect the solution accuracy if we deal with geological faults that are not aligned with the computational grid. Thus, we need to align them or use some kind of numerical strategy, e.g., by using nonfitting grids (Chaumont-Frelet et al., 2018a,b), in order to improve the accuracy.

8. Conclusions

We propose the use of refined isogeometric analysis (rIGA) discretizations for generating a massive synthetic database for deep learning inversion of 2.5D borehole electromagnetic measurements. Such a large database is essential for layer-by-layer estimation of the inverted Earth models, which may be used for real-time adjustments of the well trajectory during geosteering operations.

rIGA delivers computational savings of up to $O(p)$ compared to the high-continuity isogeometric analysis (IGA). When compared to a traditional finite element analysis (FEA) with the same mesh size and polynomial degree, rIGA provides higher improvement factors. At the same time, rIGA provides sufficiently accurate solutions for geosteering purposes.

To create a dataset for deep learning inversion, we first selected certain discretization parameters based on the results of several homogeneous solutions. Then, we checked the accuracy over homogeneous and heterogeneous media. Finally, we generated a meaningful synthetic database composed of 100,000 Earth models with the corresponding measurements in about 56 hours using a workstation equipped with two CPUs. Herein, by meaningful synthetic database we mean a set of borehole measurements that covers adequate ranges of variation of the involved variables (i.e., Earth parameterization models) and should be sufficient for a robust DL inversion.

As future work, we propose the use of artificial intelligence based techniques to perform the inversion of a large set of borehole resistivity measurements with 2D Earth models containing multiple geological faults. The required dataset for such inversion can be created synthetically and complemented by some real Earth models. To generate the synthetic database, we propose to execute the presented method over a cluster of hundreds of CPUs/threads. We will also employ more complex Earth model parameterizations including anisotropic layers.
Acknowledgment

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 777778 (MATHROCKS), the European POCTEFA 2014-2020 Project PIXIL (EFA362/19) by the European Regional Development Fund (ERDF) through the Interreg V-A Spain-France-Andorra programme, the Project of the Spanish Ministry of Science and Innovation with reference PID2019-108111RB-I00 (FEDER/AEI), the BCAM “Severo Ochoa” accreditation of excellence (SEV-2017-0718), and the Basque Government through the BERC 2018-2021 programme, the two Elkartek projects 3KIA (KK-2020/00049) and MATHEO (KK-2019-00085), the grant “Artificial Intelligence in BCAM number EXP. 2019/00432”, and the Consolidated Research Group MATHMODE (IT1294-19) given by the Department of Education. The authors also acknowledge the computer resources and the technical support provided by Barcelona Supercomputing Center through the “MareNostrum4” cluster and under the activity ID “IM-2020-3-0009”.

References


Appendix A. Computer code availability

The computer codes contributed to the research results reported in the paper as well as the varying parameters and generated synthetic dataset are publicly available on https://gitlab.com/ali.hashemian/2.5d_em. The file 2.5D_EM_parallelrun.py executes the parallel computations. We write this code in a Python 2.7 environment, which is based on Anaconda (available on https://repo.anaconda.com/archive). In this file, we set different parameters for our model problems and export them as inputs to the run-riga-em.sh file. This shell file consists of three parts. The preprocessing part that generates the computational mesh using Python. Therein we employ igakit (available on https://bitbucket.org/dalcinl/igakit) to create appropriate B-spline discretizations of the computational domain. The main part is a C code that solves the 2.5D borehole EM problem. The code needs PETSc (available on https://gitlab.com/petsc/petsc) and PetIGA (available on https://bitbucket.org/dalcinl/PetIGA) packages to be executed. It is important to note that PETSc should be configured with complex scalars, METIS, OpenMPI, ScALAPACK, and Intel MKL libraries to have a computationally efficient solution. The latter library contains PARDISO as our sparse direct solver. Finally, the third part of the shell file performs the postprocessing in Python to calculate apparent resistivities based on attenuation ratio and phase difference.