Multiscale Forward and Inverse Problems with the DGFD Method and the Deep Learning Method

by

Runren Zhang

Department of Electrical and Computer Engineering
Duke University

Date: ____________________________
Approved:

____
Qing Huo Liu, Advisor

____
William T. Joines

____
Steven A. Cummer

____
Martin A. Brooke

____
Lei Li

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2020

Abstract

Multiscale Forward and Inverse Problems with the DGFD Method and the Deep Learning Method

by

Runren Zhang

Department of Electrical and Computer Engineering
Duke University

Date: ___________________________

Approved:

____
Qing Huo Liu, Advisor

____
William T. Joines

____
Steven A. Cummer

____
Martin A. Brooke

____
Lei Li

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2020

Abstract

Fast electromagnetic (EM) forward solvers have been developed for the subsurface detection, with application includes producing synthetic logging data and instructing large-scale field test and inversion. Deep learning based full wave inversion methods have also been developed to reconstruct the underground anomaly.

Since the gas and oil industry has very high demands for the forward modeling speed when doing inversion, the inversion model is usually simplified to a 1D or 2D problem by supposing the geometry of object invariant in two or one direction. The full 3D inversion is still a hot topic for research, which requires both fast 3D forward solver and efficient inversion method. The bottleneck for the forward solver is how to solve the large-scale linear system efficiently; the bottleneck for the inversion is how to pick the global minimum from lots of local minimums efficiently for the inverse problem.

For the forward part, the domain decomposition method (DDM) inspired discontinuous Galerkin frequency domain (DGFD) method has been extended to model the vertical open borehole resistivity measurement with structured gradient meshes; besides, the DGFD method has been extended to model the logging-while-drilling (LWD) resistivity measurement in high-angle and horizontal (HA/HZ) well and curved layers with a flipped total field/scattered field (TF/SF) mixed solver. An approximated casing model has also been proposed to accelerate the large-scale curved casing modeling with borehole-to-surface measurements.

For the inversion part, a convolutional neural network based inversion has been developed to reconstruct the lateral extent and direction of the hydraulic fracture through scattered electromagnetic field data under borehole-to-surface measurements; further, the deep transfer learning is applied in the same scenario to improve the performance of the inversion. Additionally, a fully connected neural network has been developed for the Devine field data and successfully map the proppant distribution inside the hydraulic fracture with good agreement to the conventional inversion result.

This dissertation is dedicated to my beloved parents

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Abbreviations

ABC	absorbing boundary condition.
AEC	Advanced Energy Consortium.
AEM	airborne electromagnetic methods.
BiCGStab(L)	biconjugate gradient stabilized (L) method.
CEM	computational electromagnetics.
CNN	convolutional neural network.
CSEM	controlled source electromagnetic.
DDM	domain decomposition method.
DDR	deep directional resistivity.
DGFD	frequency domain discontinuous Galerkin method.
DGTD	time domain discontinuous Galerkin method.
DL	deep learning.
DoFs	degrees of freedoms.
DOI	depth of investigation.
EIT	electromagnetic impedance tomography.
EM	electromagnetics.
ERT	electromagnetic resistivity tomography.
FDM	finite difference method.
FDTD	time domain finite difference method.

FEM	finite element method.
FGMRES	flexible generalized minimal residual method.
GLL	Gauss-Lobatto-Legendre polynomial.
GPR	ground penetrating radar.
HA/HZ	high-angle and horizontal.
IE	integral equation.
ITBC	impedance transition boundary condition.
KNN	k-nearest neighbors algorithm.
LHS	left-hand-side.
LSTM	long short-term memory.
LWD	logging-while-drilling.
MAE	mean absolute error.
ML	machine learning.
MoM	method of moments.
MT	magnetotelluric methods.
OB	observation well.
PBC	periodic boundary condition.
PEC	perfect electric conductor.
PMC	perfect magnetic conductor.
PML	perfectly matched layer.
PPW	points per wavelength.
PVC	polyvinyl chloride.
QMR	quasi-minimal residual method.
RHS	right-hand-side.
RNN	recurrent neural network.
RTC	Riemann transmission condition.

SBC	scattering boundary condition.
SEM	spectral element method.
SF	scattered field.
SIE	surface integral equation.
SNR	signal-to-noise ratio.
TDIP	time-domain-induced polarization.
TF	total field.
TC	transmission condition.
TL	transfer learning.
TR	transmitter-to-receiver.
VIE	volume integral equation.

Acknowledgements

At this moment of concluding my Ph.D. study, I would like to express my most sincere gratitude to my advisor, Professor Qing Huo Liu, for his constant help and guidance through my five-year life at Duke University. His insightful suggestions and immense knowledge always help me overcome difficulties and find new paths towards challenging problems. His extreme enthusiasm, devoted attitude and open mind towards research set a model as an excellent researcher for me.

I would like to thank the committee members, Prof. William T. Joines, Prof. Steven A. Cummer, Prof. Martin A. Brooke and Prof. Lei Li for their interests in my work and the constructive suggestions on my research through all the stages from the qualify exam to the final defense. I thank them for reviewing my dissertation in their busy schedules and attending my final exam.

I also would like to thank Prof. Andrew Hilton and Prof. Genevieve Lipp to give me the opportunity to serve as the teaching assistant for ECE551 - "Programming, Data Structures, and Algorithms in C++", where I have learned a lot about programming and teaching from them and enjoyed a lot with the collaboration with them and other TA folks.

I want to thank Ms. Maria Parker and Ms. Diane Bryson from the English for International Students (EIS) team; both of my written and oral English abilities have improved a lot under their instruction.

My research work was supported by the Advanced Energy Consortium (AEC)

with ExxonMobil, Repsol, Total, and Shell as member companies and U.S. Department of Energy with National Energy Technology Laboratory award number DE-FE0031785. I am grateful to AEC's managers, advisors and researchers for their invaluable feedback and discussions, especially Dr. Mohsen Ahmadian, Dr. Alfred Kleinhammes and Dr. Douglas LaBrecque.

My internship was supported by Halliburton and Apple. I want to thank my mentors Dr. Ahmed Fouda and Dr. Ilker Capoglu at Halliburton as well as thank my mentors Dr. Zhenggang Cheng and Dr. Qianyi Zhao at Apple. Although the specific tasks were different from my Ph.D. research, I did learn a lot for different technical skill sets and achieve personal growth from the guidance of all of them.

I would like to thank Prof. Hongsheng Chen to lead me into scientific research and to teach me a lot during my graduate study. I also want to thank Prof. Shisheng Lin, Prof. Xianmin Zhang and Prof. Haogang Wang for the recommendation for my Ph.D. applications.

I appreciate the helps and instructions from my senior group members: Dr. Mengqing Yuan, Dr. Qiang Ren, Dr. Qingtao Sun, Dr. Jun Niu, Dr. Yuan Fang, Dr. Qiwei Zhan, Dr. Yunyun Hu, Dr. Weifeng Huang and Dr. Yi Zhang. Besides, I would also express special thank to my group member and housemate, Dezhi Wang. I also want to thank my junior group members: Yiqian Mao, Yongze Jia, Yang Zhong and Liangze Cui for discussions about research as well as programming techniques.

I want to show my gratitude to the visiting students in our group and neighborhood group: Zhenguan, Qiang-Ming, Mingwei, Naixing and Ke for close academic collaboration. Special thanks go to Zhen and Jiaqi as sports buddies, to Wenhao, Yi as roadtrip buddies, to Jiawei, Bingyang, Ping-Hao, Yao as moving buddies, to Jing, his parents, Yuke couple, Zheng for their delicious food. I also want to express my gratitude for the help of all visiting scholars in our group.

I would like to thank Dr. Xinyuan Zhang for the instruction on deep learning, thank Ms. Angela Chanh and Ms. Amy Kostrewa to coordinate multiple exams and applications during my Ph.D., and thank Ms. Pamela Billie to help me prepare OPT/CPT applications.

Last but not least, I thank my parents for their continuous support and enthusiasm of my Ph.D. journey, and my aunt's family's care and concern.

1

Introduction

1.1 Multiscale Forward Modeling

In computational electromagnetics (CEM) research, the prevalent techniques are finite difference method (FDM), finite element method (FEM) and method of moments (MoM). The FDM is widely used in time domain simulation due to its efficiency in time integration, but cannot model irregular geometries accurately owing to the limitation of the structured meshes; The FEM is dominating in the frequency domain simulation due to its flexibility to model complicated geometries with unstructured meshes, but requires to solve a large complex symmetric sparse matrix, which is extremely expensive when the problems involve millions of unknowns; The MoM for the surface (volume) integral equation (SIE/VIE) is based on the Green's functions, and therefore can achieve high accuracy for radiation and scattering problems by only discretizing the boundary elements (the scattering objects). However, the MoM is limited to the problems that Green's functions are available, typically homogeneous or layered background and therefore is not as general as the FDM and FEM.

For multiscale problems, the dimension of different objects can be dramatically different. For hydraulic fracturing, the fracture can be a very thin sheet with thickness of millimeters but transversal dimension of hundred meters; the radius of open or cased boreholes are around centimeters, but the length of them can be more than 1 km; the background formation layers may be curved and of different conductivity. To tackle with this problem, the FDM will lose the accuracy due to curved objects and the MoM will also be not accurate due to the approximation of the Green's function. In contrast, the FEM is more suitable for this problem, with the help of some novel techniques like the domain decomposition method (DDM).

In the first part of the dissertation, I will extend the DDM inspired discontinuous Galerkin frequency domain (DGFD) method to model multiscale problems in frequency domain, which hybridizes the spectral element method (SEM) and the FEM. As a branch of the FEM, the SEM [Lee and Liu (2007)] employs hexahedral elements to discretize the computational domain and achieves spectral accuracy with arbitrary high-order basis functions. Based on the Gauss-Lobatto-Legendre (GLL) integral points, both the elemental mass matrices and elemental stiffness matrices can be sparse for the high-order basis functions due to the GLL points-based Lagrangian polynomials' orthogonality. Moreover, the higher-order basis functions the SEM employs, the fewer meshes and fewer degrees of freedom (DoFs) it requires for the same accuracy. When modeling multiscale problems, the whole computation domain is decomposed into several subdomains according to their geometry characteristics to balance the accuracy and efficiency. The irregular structures are generally modeled by the dense unstructured meshes with low-order basis functions to capture the detailed shapes; the large-scale background is generally modeled by the coarse structured meshes with high-order basis functions to save the DoFs. By following this spirit, a hierarchical sudoku mesh has been proposed for the vertical well logging problem to reduce the memory consumption [Zhang et al. (2019a)]. To couple different subdomains together, the numerical flux should be calculated correctly. In this dissertation, the Riemann transmission condition (RTC) has been employed to ensure different subdomains are cemented accurately [Sun et al. (2017)]. The RTC originates from the upwind flux scheme in the discontinuous Galerkin time domain (DGTD) method [Cockburn et al. (2000)] and appears a little different in frequency domain, like a combination of a correctly modeled numerical flux term and the first-order transmission condition [Zhao et al. (2007)]. Therefore, this type of DDM can be called the DGFD method. The numerical validations show that the RTC can accelerate the computation while guaranteeing the required accuracy [Sun (2017)].

However, the SEM encounters more serious dipole source singularity problem than the FEM, since the high-order basis functions inside one element cannot alleviate the singularity impact, but the first derivative discontinuity at the element interface can. Therefore, if we are interested in the field distribution near the source, several elements are required between the sources and receivers to reduce the source singularity effect. The extra element requirement for the source region may induce a large amount of unnecessary DoFs, especially for the logging-while-drilling (LWD) problems since the source is moving and all places the source reaches require denser meshes. To solve the LWD problem elegantly, a flipped total field/scattered field (TF/SF) DGFD solver has been proposed [Zhang et al. (2019b)] by extending the mixed TF/SF DGFD solver [Sun et al. (2018)] to distribute the incident fields of the dipole source to the interface between the source subdomain and source free subdomain.

Besides, to model extremely thin structures like hydraulic fractures and casing walls efficiently without dense meshes in the thickness direction, the impedance transition boundary condition (ITBC) [Mitzner (1968)] is employed to treat the thin structure as a surface and only discretize along the transversal dimension. For the large-scale curved casing problem, the ITBC can help to reduce up to 19/20 unknowns. To further reduce the unknowns and accelerate the solution phase of the large-scale curved casing, an approximated casing model has been proposed to reduce up to 9/10 meshes with a limited accuracy loss [Zhang et al. (2020)].

1.2 Deep Learning Inversion

One of the most important applications of the forward modeling is to solve the inverse problem [Calvetti and Somersalo (2018)], which is usually ill-posed due to the insufficient data to identify the cause unambiguously. A traditional method to solve it is the regularization-based algorithm [Tikhonov (1963)], which replaces the ill-posed problem by a nearby well-posed one; another family of methods is based on the Bayesian methodology [Tarantola (1987)], which can incorporate extra information to supplement the noisy data by modeling unknowns as random variables to emphasize the uncertainty about its value. Both approaches can then mathematically be viewed as a optimization problem [Fletcher (2013)]. To solve the optimization problem, again there are two directions. One classical optimization method calculates the first and/or second derivatives of the cost function in terms of the unknowns and employs iterative optimization algorithms to repetitively call forward solvers. However, it only guarantees a local minimum instead of the desired global one. Another optimization method family is zero-order optimization methods or ”metaheuristics” [De Castro (2006)], for example genetic or evolutionary algorithms. They are generally iterative stochastic algorithms, progressing toward the global extreme of the objective function through sampling. Though effective in some cases, they cannot always be characterized by convergence theorems and computational heavy for large-scale problems.

Due to the great success of applying deep convolution neural networks (CNNs) on object classification and segmentation tasks, they have also been employed to improve the resolution of inverse problems in imaging [McCann et al. (2017)] such as denoising, deconvolution, super-resolution, and medical image reconstruction. Taking the advance in medical image reconstruction as example, one of highest cited paper [Jin et al. (2017)] applied a CNN-based method to the X-ray CT, which showed improvement for sparse-view reconstruction by substituting the regularization-based iterative inversion with a CNN. However, the CNN still needed the filtered back projection as a direct inversion solver to preprocess the sinogram to guarantee a good signal-to-noise ratio (SNR). The similar application in geophysics originally lies in the ground penetrating radar (GPR) society, where the CNN is used for image segmentation [Travassos et al. (2018)], as well as in the exploration geophysics society, where the CNN is used to extract features from 3D seismic volumes [Araya-Polo et al. (2017)]. In recent two years, some pioneering work has been done towards the EM geophysical inversion. For example, a deep NN based on fully convolutional architecture has been trained by large synthetic datasets in [Puzyrev (2019)]. The performance of the method was demonstrated on models of strong practical relevance representing an onshore controlled source electromagnetic (CSEM) CO2 monitoring scenario. The pre-trained networks can reliably estimate the position and lateral dimensions of the anomalies, as well as their resistivity properties.

Another family of inverse problems in subsurface detection is well logging interpretation. The presence of thin beds, anisotropy, and different borehole trajectory can make raw and apparent resistivity logs abnormal [Pardo and Torres-Verdín (2015)], which cannot be interpreted without corrections. Apparent resistivity "horns" and abnormal separation of multi-resolution resistivity curves are two common situations of interpretation. Without numerical simulation, it becomes impossible to detect, quantify, and reduce those effects to reliably assess hydrocarbon saturation. Therefore, the efficient simulation methods and the robust inversion procedures are required to produce practical, efficient, and reliable automatic inferences of true layer resistivities and permittivities and/or determine the bed boundary [Wang et al. (2018)] in the presence of geometric and other environmental effects.

The use of neural networks (NN) in well logging has been popular for nearly two decade [Nikravesh (2007)]. Many successful applications have been documented, including the use of NNs for predicting permeability from well logs [Bruce et al. (2000)]. In this application, a Bayesian NN was used as a nonlinear regression tool to develop transformation between well logs and core permeability. The same tool can also be applied to estimate porosity and fluid saturations. Another important application is the clustering of well logs for the recognition of lithofacies [Rogers et al. (1992)], which provides useful information for improved petrophysical estimates and well correlation.

Recently, a CNN and recurrent neural network (RNN) combined method [Shahriari et al. (2018)] was proposed to map the conductivity of a 1D layered subsurface model from the borehole resistivity measurement in real time, to assist the geosteering application in LWD. A specific class of RNN, the Long Short-Term Memory (LSTM) Network was first employed to reduce the dimensionality of the input measurements; then a series of 1D CNN based on a modified residual block was connected with interleaved pooling operators; at last, the fully connected layer mapped the CNN's results into the space of subsurface resistivity properties. The significant acceleration over all other conventional inversion methods made the machine learning (ML)-based well logging inversion promising; however, the reported results can only indicate the formation to some extent of large errors. On one side, the extracted property needs to be more representative; on the other side, the faster 2D or 3D forward solvers need to be involved to produce a massive number of training data.

The second part of the dissertation will focus on reconstructing the hydraulic fracture with a deep learning (DL)-based inversion, including two scenarios. The first scenario is a follow-up work for the large-scale curved casing, where the vertical fracture's direction and dimension have been estimated by a CNN. Two different training strategies have been proposed to train the network, where one is purely using the approximated casing data and the other one is mixing a few true casing data to the approximated casing data. Furthermore, a deep transfer network is applied to provide another approach to reconstruct the fracture. A relatively large amount of approximated casing data is employed to train a CNN, which is then adapted to true casing data by fine-tuning top layers. The second scenario is to reconstruct proppant distribution inside a horizontal fracture from the Devine field data. The proppant distribution is first parameterized by using the B-spline closed curve with 16 control points. The fast EM solver is then employed to produce the synthetic data to train a fully connected NN.

To begin with, the numerical methods for subsurface problems have been reviewed in Chapter 2, including the SEM, the ITBC, the SF solver, the RTC, the DDM and the TF/SF mixed solver. The hierarchical sudoku mesh and the flipped TF/SF mixed solver have been proposed in Chapter 3 to save DoFs for the vertical well logging and the LWD problem; one cavity identification case and one large-scale oil trap case including a salt dome, a fold and a fault have been showcased to demonstrate their advantages. The approximated casing model is then proposed and used to generate the approximated casing data to train the CNN to reconstruct the hydraulic fracture's direction and dimension in Chapter 4; a deep transfer learning is also developed and applied to the same problem. In Chapter 5, a fully connected NN has been developed to reconstruct the fracture shape from the Devine field data.

2

Review of Numerical Methods for Subsurface Problems

In subsurface detection area, EM methods play an important role [Chen et al. (2016)]: 1. Borehole EM methods [Doll et al. (1949)] are an important part in well logging. Since water is conductive while hydrocarbons (oil and gas) are insulated, resistivity measurements are good indicators of the presence of hydrocarbon-bearing zones. On the other hand, water has a high dielectric constant, and propagation resistivity measurement is therefore a good sensor of moisture content. 2. GPR [Daniels (1996)] has been routinely used in geological surveying, especially to obtain high-resolution near-surface imaging for geological, archeology, environmental (including demining), geotechnical, and urban planning applications. 3. Magnetotelluric (MT) methods [Chave and Jones (2012)] are to use natural EM fields to investigate the electrical conductivity structure of the earth, with frequencies from 0.001 Hz to 10^4 Hz and investigation depth from 50 m to several kilometers. The shortcomings are limited resolution and low SNR. 4. Airborne EM methods (AEM) [Ward et al. (1988)] are widely used in geological surveys and prospecting for conductive ore bodies. These methods are suitable for large area surveys because of their speed and cost effectiveness. In contrast to ground EM methods, airborne EM methods are usually used to outline large-scale structures while ground EM methods are preferred for more detailed investigations. 5. Other EM methods include: Inductive EM method [Keller and Frischknecht (1966)], time-domain EM method [Danielsen et al. (2003)] and CSEM [Constable (2010)].

In this dissertation, I will focus on the numerical models for borehole EM methods. When numerically modeling the borehole measurements, one first needs to balance the fidelity in the approximation of the tool's physical geometry, transmission/reception scenario, geophysical environment's characteristics (inhomogeneity, anisotropy, and dispersion), expected precision, required runtime, and available computation resources, and then decide to include or neglect the following: borehole, drilling fluid and invasion zone, casing, finite antenna geometry, mandrel, transient effects in a narrow-band transmission, material anisotropy, two or three-dimensional geophysical inhomogeneities, and so on [Chen et al. (2016)].

The general comparison between different numeric techniques has been given in Section 1 and the conclusion is that the FEM is the most general solver to handle complicated and multiscale models. The recent advances of the FEM mainly focus on more efficient Krylov subspace-based iterative solvers [Saad (2003)] and corresponding preconditioners, high performance direct solvers [Davis et al. (2016)], nonconformal meshes/DDM [Lee and Peng (2017)], adaptive meshes [Paszyński et al. (2010)], and parallel computation [Puzyrev et al. (2013)]. For borehole EM forward solvers, a DDM [Ma et al. (2014b)] is employed to design the tool more efficiently by meshing the tool with dense hexahedron meshes and background formation with coarse hexahedron meshes; another DDM [Chen (2017)] is employed to model a 2D layered medium with borehole, where the layer with irregular borehole washout or nonuniform invasion of drilling fluid can be solved with conventional FEM and the layer with longitudinal homogeneity can be efficiently solved by the semi-analytical

FEM; a 2.5 D FEM [Chaumont-Frelet et al. (2018)] is employed with non-fitting mesh for the LWD to release the meshing burden for the FEM.

The DGFD method combines the FEM and the SEM through the DDM and has been reviewed in this chapter to demonstrate its basic idea and characteristic. This chapter is organized as follows: the basic concept and formula for the SEM has been reviewed in Section 2.1; I then discuss how to determine the computational region in Section 2.2; a special boundary condition ITBC and a scattered field (SF) solver have been reviewed in Section 2.3 and Section 2.4 to model the fracture and to remove the source singularity, respectively. Finally, the DGFD method has been reviewed, including the Riemann transmission condition (RTC), the work flow and the TF/SF mixed solver respectively in Section 2.5, Section 2.6 and Section 2.7, .

2.1 Spectral Element Method

The frequency domain SEM method starts from the vector Helmholtz equation for electric fields

$$\frac{1}{j\omega\mu_0}\nabla\times(\tilde{\mu}_r^{-1}\nabla\times\mathbf{E})+j\omega\epsilon_0\tilde{\epsilon}_r\mathbf{E}=-\mathbf{J}-\frac{1}{j\omega\mu_0}\nabla\times(\tilde{\mu}_r^{-1}\mathbf{M}) \tag{2.1}$$

where the complex relative permeability $\tilde{\mu}_r=\mu_r+\frac{\sigma_m}{j\omega\mu_0}$ and the complex relative permittivity $\tilde{\epsilon}_r=\epsilon_r+\frac{\sigma_e}{j\omega\epsilon_0}$.

Employ the edge basis functions $\mathbf{\Phi}$ to test (2.1), we can get the weak form of the Helmholtz equation as

$$\begin{aligned}\frac{1}{j\omega\mu_0}\int_\Omega(\nabla\times\mathbf{\Phi})\cdot(\tilde{\mu}_r^{-1}\nabla\times\mathbf{E})dV+j\omega\epsilon_0\int_\Omega\mathbf{\Phi}\cdot\tilde{\epsilon}_r\mathbf{E}dV-\int_{\partial\Omega}\mathbf{\Phi}\cdot(\hat{n}\times\mathbf{H})dS\\=-\int_\Omega\mathbf{\Phi}\cdot\mathbf{J}dV-\frac{1}{j\omega\mu_0}\int_\Omega(\nabla\times\mathbf{\Phi})\cdot(\tilde{\mu}_r^{-1}\mathbf{M})dV\end{aligned} \tag{2.2}$$

The (2.2) includes one integral by parts for the second-order derivative term and one integral by parts for the magnetic source $\mathbf{M}$. The second is correct when the magnetic source does not contact the boundary.

To evaluate the boundary integral in (2.2), we generally employ three equations for perfect electric conductor (PEC), perfect magnetic conductor (PMC) and absorbing boundary condition (ABC), respectively.

$$\hat{n} \times \mathbf{E} = 0 \quad \text{on } \Gamma_{\text{PEC}} \tag{2.3}$$

$$\hat{n} \times \mathbf{H} = 0 \quad \text{on } \Gamma_{\text{PMC}} \tag{2.4}$$

$$\lim_{r->\infty}(r\nabla \times \mathbf{H} + jk\mathbf{r} \times \mathbf{E}) = 0 \quad \text{on } \Gamma_{\text{ABC}} \tag{2.5}$$

For the electric field-based Helmholtz equation, both the PEC and PMC have zero boundary condition while the PEC does not require the boundary tangential basis functions at all; therefore, the PEC is the most efficient boundary condition when computing.

In this dissertation, the high-order accuracy of the SEM is achieved in use of the Lagrangian polynomials [Lee et al. (2006)] and the tangential continuity of the electric field is taken care by the vector basis functions [Peterson et al. (1998)]. The elemental mass and stiffness matrices generated from the exact integral will be full matrices, which means the number of the nonzero entries is quadratic to the cube of the basis function order (N^3) and usually linear to the element number. Since the number of nonzero entries is critical to the runtime of the final linear system solver, the quadratic growth of the nonzero entries is generally unacceptable. To reduce the quadratic growth to the super-linear growth, the GLL type Lagrangian polynomial

$$\phi_m^{(M)}(x) = \frac{-(1-x^2)L_N'(x)}{N(N+1)L_N(x_m)(x-x_m)}, m = 0, ..., M \tag{2.6}$$

is employed to interpolate the testing and basis functions, where $L_N(x)$ is the Legendre polynomial of degree N and x_m are the GLL points. If the integration also employs the Gauss-Lobatto quadratures that collocates with the governing points of the basis (testing) functions, the elemental mass matrix will have N nonzero entries per row and the stiffness matrix will have $4\mathrm{N}^2 + 3\mathrm{N}$ nonzero entries per row for a N^{th}-order basis function, while the column number is $3\mathrm{N}^3+6\mathrm{N}^2+3\mathrm{N}$. Therefore, the number of the nonzero entries is of order $\frac{5}{3}$ to the cube of the basis function order. The sparsity of the element matrix is $\frac{4}{3(\mathrm{N}+1)} - \frac{1}{3(\mathrm{N}+1)^2}$. However, the conclusion is only true for structured mesh; for unstructured mesh, due to the Jacobian matrix, the elemental matrix may not be sparse as expected. Therefore, the high-order basis function can be more effective for the structured mesh.

2.2 Computational Boundary

Most geophysics problems are open boundary problems and therefore how to choose the computational region size and boundary condition is critical. For the wave problems, the ABC is generally required because the outgoing waves cannot be absorbed by either the PEC or the PMC. However, the geophysics problems are more similar to diffusion problems than wave problems, because the relative permittivity $\tilde{\epsilon}_r$ is nearly pure imaginary given low frequency and high conductivity. For the diffusion like problems, the fields may decay very rapidly from the source region; therefore, the ABC may not be necessary sometimes. Instead, a PEC or PMC boundary can be used to truncate the domain and the boundary can be placed where the field intensity is 10 percent of the field intensity at the receiver region to guarantee a 1 percent error. The incident field for homogeneous or layered media can be easily obtained by analytical solution, while the scattered field may need some approximation.

Let's use a simplified experimental setup in Devine [Ahmadian et al. (2018)] to illustrate the methodology. The diagram is shown in Fig. 2.1: One 5-mm thick, 25-m radius circular fracture of $\epsilon_r = 10^7$, $\mu_r = 1$, $\sigma = 100$ S/m, locates horizontally at $z = -60$ m. The background formation is supposed homogeneous of $\epsilon_r = 10^6$, $\mu_r = 1$, $\sigma = 0.02$ S/m. Monitors 1 to 4 are four vertical observation wells with position

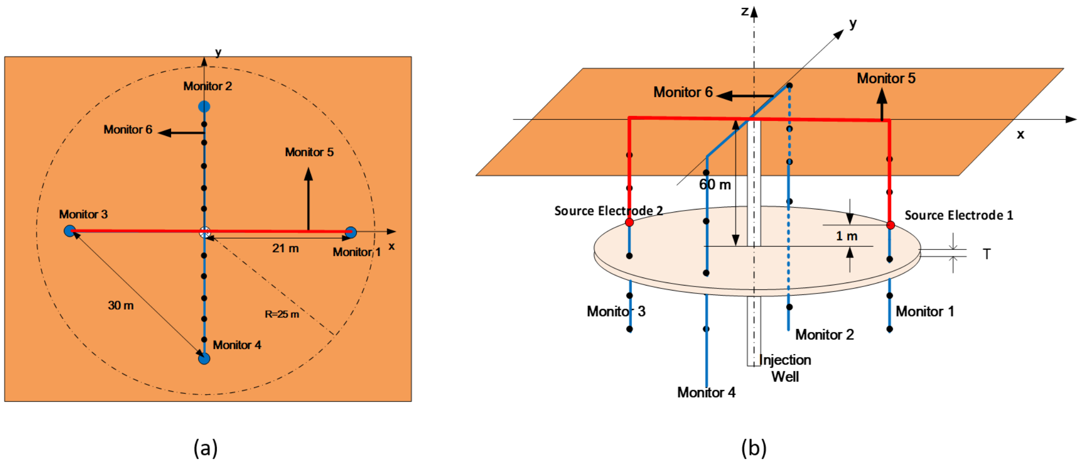


FIGURE 2.1: Diagram of the Devine case

(21,0,-100∼0) m, (0,21,-100∼0) m, (-21,0,-100∼0) m, (0,-21,-100∼0) m, respectively; Monitors 5 and 6 are two horizontal observation wells with position (-21∼21,0,0) m, (0,-21∼21,0) m. The positive electrode and negative electrode are put at (21,0,-59) m, (-21,0,-59) m with a horizontal direct connecting cable. The working frequency is 1 Hz.

By calculating the complex relative permittivity $\tilde{\epsilon}_r = 10^6 + \frac{0.02}{j2\pi\times1\times8.85\times10^{-12}} = 10^6 + j3.60\times10^8$, we can find that the real part is smaller than the imaginary part as expected and this is a diffusion like problem. Therefore we can use the analytical solution to find the position to truncate the computational domain. The normalized field distribution contour (log scale) on a $x-z$ plane at $y = 21$ m is plotted in Fig. 2.2; the $-100 \sim 0$ m portion of the middle vertical line will correspond to the Monitor 2.

Notice that the relative field intensity around the Monitor 2 is above -20 dB for the main component E_x; to ensure a -40 dB attenuation between the reflected field and the incident field, we can put PEC boundaries at $x, y = \pm 200$ m and $z = -235, 115$ m, where the relative field intensity is between -40 dB and -50 dB. Compare the determined computational domain to both the wavelength $\lambda_0 = 300$ km and the skin

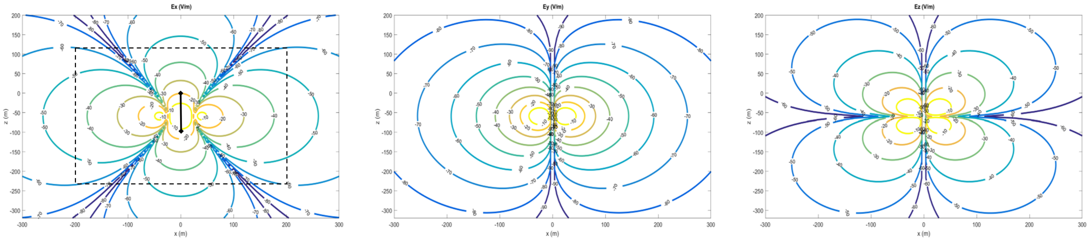


FIGURE 2.2: Relative field intensity contour at a vertical plane in log scale

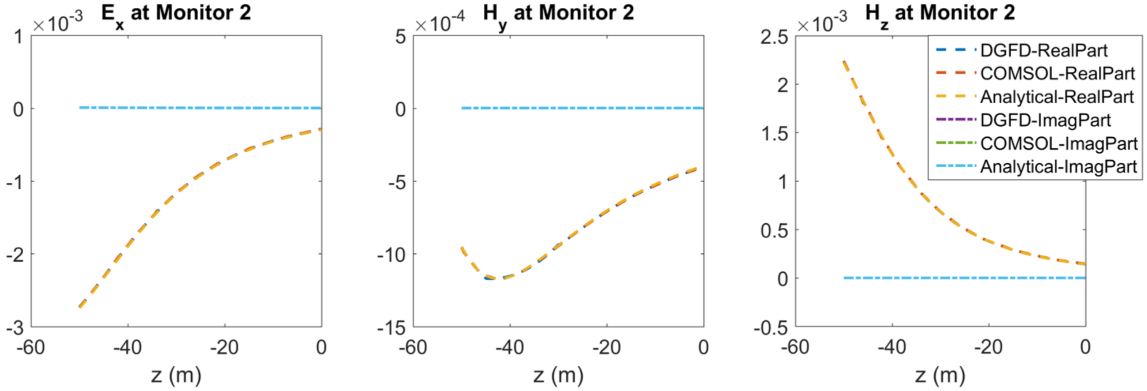


FIGURE 2.3: Main component for the incident fields along with the Monitor 2

depth $\delta = 3.56$ km, we can find the domain size is only about 10 percent of the skin depth, which indicates that when the source is involved in the computational domain, the fields actually decay faster than that the skin depth implies, because the skin depth is more meaningful for the plane wave.

By using gradient meshes ranging from 15 m to 50 m, the above PEC truncated domain can be discretized into 1404 hexahedrons with the third-order basis functions and then generate a final linear matrix of 108,276 DoFs with 0.64% error compared to the analytical solution. To achieve the same level of accuracy, the COMSOL will require about 120,000 tetrahedrons and 750,000 DoFs. The field distribution on Monitor 2 from $z = -50$ m to $z = 0$ m are compared in Fig. 2.3 for three main components $\mathrm{E_x}$, $\mathrm{H_y}$ and $\mathrm{H_z}$.

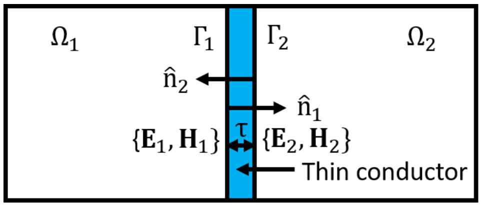


FIGURE 2.4: Diagram for the thin layer modeled by the ITBC

2.3 Impedance Transition Boundary Condition

Hydraulic fracturing is one important technique to exploit subsurface fossil resources like oil and natural gas as well as geothermal energy. The generated fractures can provide efficient seepage for fluid to flow through the earth formation. To monitor the hydraulic fractures, EM survey is considered as one of the effective geophysical tools with the aid of high conductivity proppants inside the fracture. Several numerical methods are applied to analyze the EM responses of fractures under different geophysical environments. The integral equation (IE) is prevalent [Yang et al. (2015)] due to its high accuracy and small computation domain, especially with the introduction of thin dielectric sheet-based surface integral equation method for fracture modeling [Ren et al. (2016)]. However, the IE methods are limited to scenarios that the Green's functions are available and not suitable for complicated geophysical environments, where the FEM is more general [Weiss et al. (2016)].

In this section, we introduce the ITBC [Mitzner (1968)] to approximate the effect of thin imperfect conductor or dielectric to surrounding media as depicted in Fig. 2.4 for the SEM; the ITBC simplifies the 3D interaction among the fields around the thin layer into a 2D interaction between the tangential fields at two sides of the thin layer. The basic assumption for the ITBC is that inside the thin layer, the fields vary much faster in the vertical direction than the transverse directions, which degenerates the 3D source-free Helmholtz equation into the 1D source-free Helmholtz equation as follows,

$$\frac{\partial^2 \mathbf{E}}{\partial z^2} + k^2 \mathbf{E} = 0 \tag{2.7}$$

with the solution

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{+}(x,y)e^{-jkz} + \mathbf{E}^{-}(x,y)e^{+jkz} \tag{2.8}$$

$$\mathbf{H}(\mathbf{r}) = \mathbf{H}^{+}(x,y)e^{-jkz} - \mathbf{H}^{-}(x,y)e^{+jkz} \tag{2.9}$$

where $\mathbf{E}^{+}(x,y)$ represents the electric field distribution of (x,y) for the positive propagating waves and $\mathbf{E}^{-}(x,y)$ represents the electric field distribution of (x,y) for the negative propagating waves. On the other hand, the 1D Maxwell equation can also be discretized as

$$\hat{n}_1 \times (\mathbf{H}_2 - \mathbf{H}_1) = j\omega\epsilon\tau_{eq}\hat{n}_1 \times (\mathbf{E}_1 + \mathbf{E}_2) \times \hat{n}_1 \tag{2.10}$$

$$\hat{n}_1 \times (\mathbf{E}_2 - \mathbf{E}_1) = -j\omega\mu\tau_{eq}\hat{n}_1 \times (\mathbf{H}_1 + \mathbf{H}_2) \times \hat{n}_1 \tag{2.11}$$

for the three layer model as in Fig. 2.4. When considering the reflection and transmission at interfaces Γ_1 and Γ_2, the thickness of the thin layer τ should be updated to τ_{eq} [Karlsson (2009)]

$$\tau_{eq} = \frac{e^{-jk\tau} + e^{+jk\tau} - 2}{-jk(e^{-jk\tau} - e^{jk\tau})} = \frac{1}{k}\tan(\frac{k\tau}{2}) \tag{2.12}$$

by substituting the $\mathbf{E}_1, \mathbf{E}_2, \mathbf{H}_1, \mathbf{H}_2$ from (2.9), (2.8) into (2.10), (2.11). A matrix representation for the ITBC will be then

$$\begin{bmatrix} \hat{n}_1 \times \mathbf{E}_2 \times \hat{n}_1 \\ \hat{n}_1 \times \mathbf{H}_2 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \hat{n}_1 \times \mathbf{E}_1 \times \hat{n}_1 \\ \hat{n}_1 \times \mathbf{H}_1 \end{bmatrix} \tag{2.13}$$

where

$$\mathbf{T} = \frac{1}{1 + k^2\tau_{eq}^2} \begin{bmatrix} 1 - k^2\tau_{eq}^2 & 2j\omega\mu\tau_{eq} \\ 2j\omega\epsilon\tau_{eq} & 1 - k^2\tau_{eq}^2 \end{bmatrix} \tag{2.14}$$

Similar to other boundary condition, the ITBC is also combined to the SEM through the boundary integral term [Sun et al. (2017)] and therefore it is more straightforward if we rearrange the $(\hat{n}_1 \times \mathbf{H}_1), (\hat{n}_2 \times \mathbf{H}_2)$ as the vector at the left hand side as

$$\begin{bmatrix} \hat{n}_1 \times \mathbf{H}_1 \\ \hat{n}_2 \times \mathbf{H}_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} \hat{n}_1 \times \mathbf{E}_1 \times \hat{n}_1 \\ \hat{n}_2 \times \mathbf{E}_2 \times \hat{n}_2 \end{bmatrix} \tag{2.15}$$

where the entries of the admittance matrix are

$$Y_{11} = Y_{22} = \frac{Z_{11}}{Z_{11}^2 - Z_{12}^2}, \quad Y_{12} = Y_{21} = -\frac{Z_{12}}{Z_{11}^2 - Z_{12}^2} \tag{2.16}$$

$$Z_{11} = \frac{j\omega\mu}{k\tan(k\tau)}, \quad Z_{12} = \frac{j\omega\mu}{k\sin(k\tau)} \tag{2.17}$$

By incorporating the ITBC into (2.2), we can get the updated weak form

$$\frac{1}{j\omega\mu_0}\int_\Omega (\nabla \times \mathbf{\Phi}) \cdot (\tilde{\mu}_r^{-1}\nabla \times \mathbf{E})dV + j\omega\epsilon_0\int_\Omega \mathbf{\Phi} \cdot \tilde{\epsilon}_r\mathbf{E}dV - \int_{\partial\Omega} \mathbf{\Phi} \cdot (\hat{n} \times \mathbf{H})dS$$

$$-\int_{\Gamma_1} \mathbf{\Phi}\cdot(Y_{11}\hat{n}_1\times\mathbf{E}_1\times\hat{n}_1+Y_{12}\hat{n}_2\times\mathbf{E}_2\times\hat{n}_2)dS-\int_{\Gamma_2} \mathbf{\Phi}\cdot(Y_{21}\hat{n}_1\times\mathbf{E}_1\times\hat{n}_1+Y_{22}\hat{n}_2\times\mathbf{E}_2\times\hat{n}_2)dS$$

$$= -\int_\Omega \mathbf{\Phi} \cdot \mathbf{J}dV - \frac{1}{j\omega\mu_0}\int_\Omega (\nabla \times \mathbf{\Phi}) \cdot (\tilde{\mu}_r^{-1}\mathbf{M})dV \tag{2.18}$$

It is worthwhile to notice that the derivation until now only excludes the thin layer from the computational domain but keep the distance between two interfaces Γ_1 and Γ_2 unchanged. However, when implementing the ITBC, the structure is usually replaced by coplanar interfaces. This simplification can induce a redundant phase difference and amplitude decay due to the extra traveling distance in the background. Fortunately, for the fracture modeling, the side effect is usually negligible. To validate the ITBC's accuracy, we continue the Devine case study with the same setup but use the ITBC to model the fracture; the normalized scattered fields from the fracture are compared in Fig. 2.5. The relative errors between the DGFD and COMSOL are 0.9% for scattered electric fields and 1.3% for scattered magnetic fields.

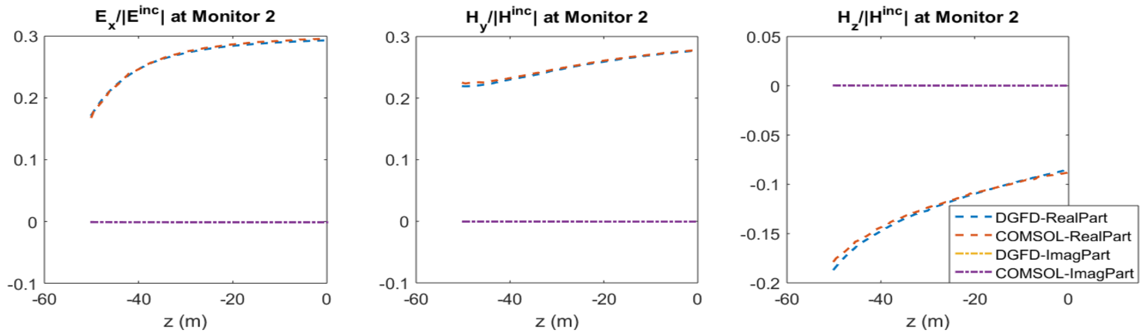


FIGURE 2.5: Main component for the scattered fields along with the Monitor 2

2.4 Scattered Field Solver

To review the source singularity problem, a salt dome case is reviewed as shown in Fig. 2.6. One 2-m thick fan shape salt dome with inner radius $a_1 = 5$ m and outer radius $a_2 = 10$ m locates horizontally at $z = 0$ m. Its conductivity $\sigma_s = 0.001$ S/m. The background formation where the salt dome resides is of conductivity $\sigma_{b2} = 0.01$ S/m and thickness 2 m. The conductivity of the top layer formation is $\sigma_{b1} = 0.02$ S/m and of the bottom layer formation is $\sigma_{b3} = 0.005$ S/m. The borehole is of radius $r_{bh} = 0.5$ m and of conductivity $\sigma_{bh} = 0.1$ S/m. The permittivity and permeability of all the materials mentioned above are $\epsilon_r = 80$ and $\mu_r = 1$. A 25 kHz magnetic dipole polarized at z direction is put at (0.01,0,-25) m and the receivers are put at (0.01,0,-20 to 20) m.

By employing the method discussed in Section 2.2, the computational domain is determined at -100 m $\leqslant x \leqslant 100$ m, -100 m $\leqslant y \leqslant 100$ m, -101 m $\leqslant z \leqslant 101$ m with PEC truncation. To show the source singularity problem, we can use the homogeneous background with middle layer property to show the source singularity. For example, even if we use the gradient mesh (total element number 4,619) as shown in the Fig. 2.7 with the z-direction mesh size 2.5 m around the source, the main component $\mathrm{H_z}$ in Fig. 2.8 shows strong variation inside the -21 m $\leqslant z \leqslant -18.5$ m element. The reason is that the problematic element is only one element away from

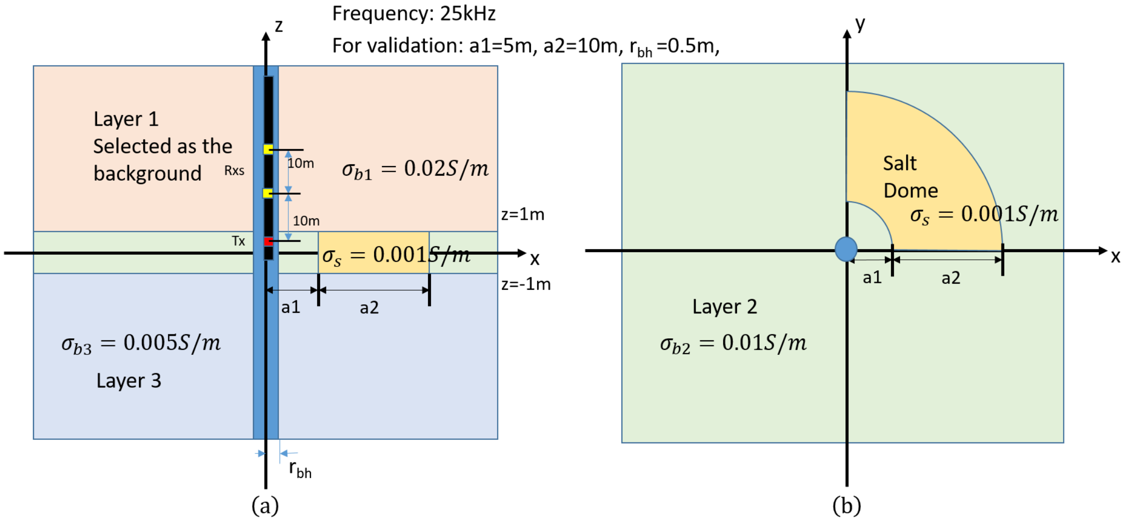


Figure 2.6: Diagram of the salt dome case

the source element. One straight forward but costly approach is to increase the z-direction density of meshes around the source as shown in Fig. 2.9, which increases the total element number to 6,109. Consequently, the main component in Fig. 2.10 shows good agreement as expected, since the closest element for the field plotting is now three elements away from the source element. Our experience shows that for the magnetic dipole, in the direction along with the polarization direction, the source singularity require at least two elements to disappear. For convenience, we denote the region where the field is erroneous due to the source singularity as the source singularity region. Notice that not only the receivers, but also the domain decomposition cutting face should not be put inside the source singularity region.

Since the singularity is caused by the imposed dipole source, we can extract the imposed source from the governing equation and solve the scattered fields instead. The source for the SF equation will then be the equivalent sources from the objects different from the background. The SF equation only modifies the source term of

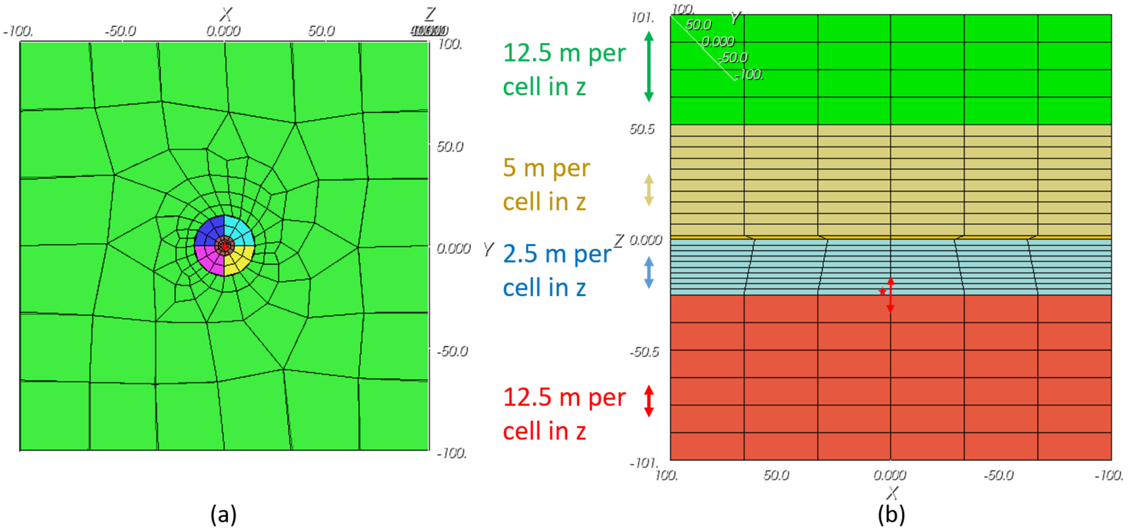


FIGURE 2.7: Relative coarse meshes of the salt dome case to show the source singularity

the (2.1) as

$$\frac{1}{j\omega\mu_0}\nabla\times(\tilde{\mu}_r^{-1}\nabla\times\mathbf{E}_s)+j\omega\epsilon_0\tilde{\epsilon}_r\mathbf{E}_s=-j\omega\epsilon_0(\tilde{\epsilon}_r-\tilde{\epsilon}_{rb})\mathbf{E}_b-\nabla\times(\mathbf{I}-\tilde{\mu}_r^{-1}\tilde{\mu}_{rb})\mathbf{H}_b \quad (2.19)$$

Instead of only integrating over the elements that contain sources, the SF equation requires us to integrate over all elements with material properties different from

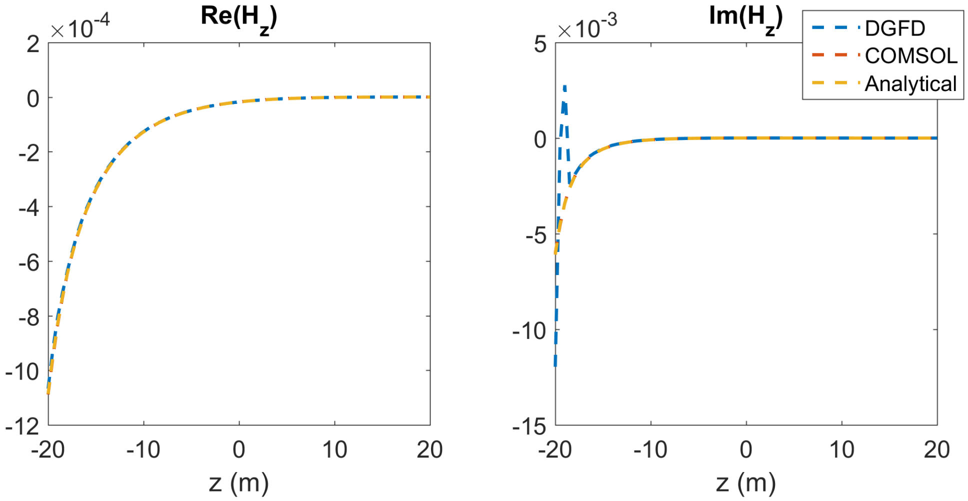
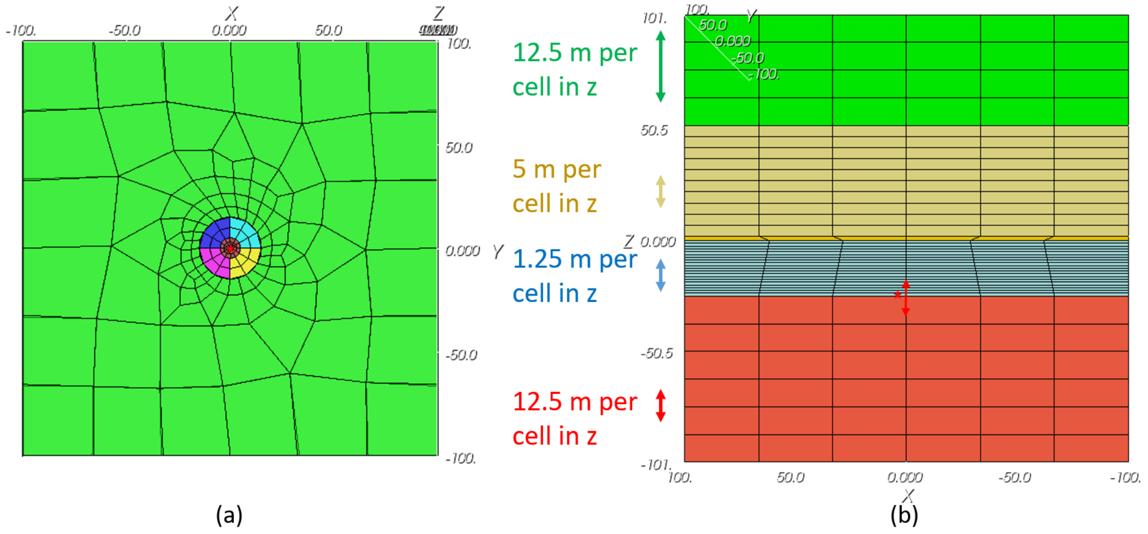


FIGURE 2.8: Main component H_z to show the source singularity



(a)

(b)

FIGURE 2.9: Relative dense meshes of the salt dome case to remove the source singularity

the background. For layered background cases, like the salt dome case, it can be approximately considered that most elements in the computational region are integrated over twice; one for the system matrix assembling, and the other for the source term assembling. If the source is fixed, then the time complexity for assembling the linear equation is still linear to the element number. However, if the source is mov-

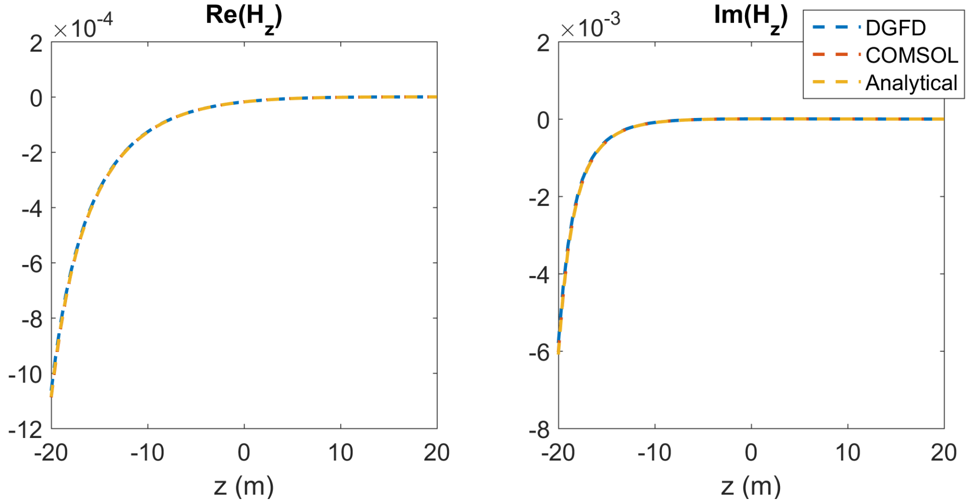


FIGURE 2.10: Main component H_z to show the removal of the source singularity

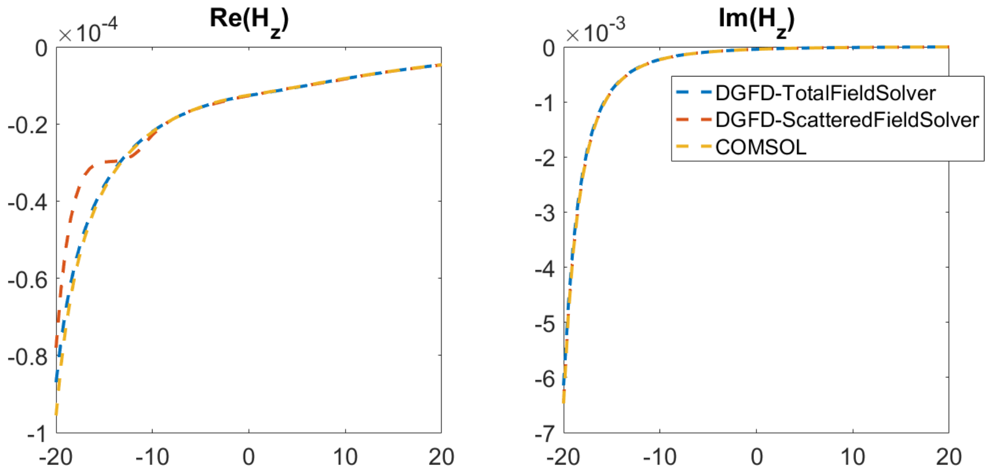


FIGURE 2.11: Main component H_z of the incident field

ing, then the time complexity for assembling the linear equation will be dominated by the source term and bilinear to both the number of elements and logging steps. A elegant solution will be discussed in the Section 3.2.

Let's come back to the salt dome case. With the help of the SF solver and select the top formation layer as the background material, the incident field for the layered medium with no salt dome is computed as shown in Fig. 2.11 for 10-m homogeneous meshes in z direction, except the 2-m meshes for the middle layer as shown in Fig. 2.12. The real part shows some errors around $-20 \text{ m} \leqslant z \leqslant -10 \text{ m}$, but the relative error for the magnetic field is still as small as 0.8%, because the real part is much smaller than its imaginary part. The error of the real part is because the mesh is too big compared to the 10-m skin depth (borehole) to capture the smaller real part. The total element number therefore reduces to 3,129. As expected, increasing meshes at the erroneous region can eliminate the error but make the total unknowns larger. For application, since generally it is hard to detect a smaller component like the real part of magnetic field in this application, it is more reasonable to use these meshes than denser meshes.

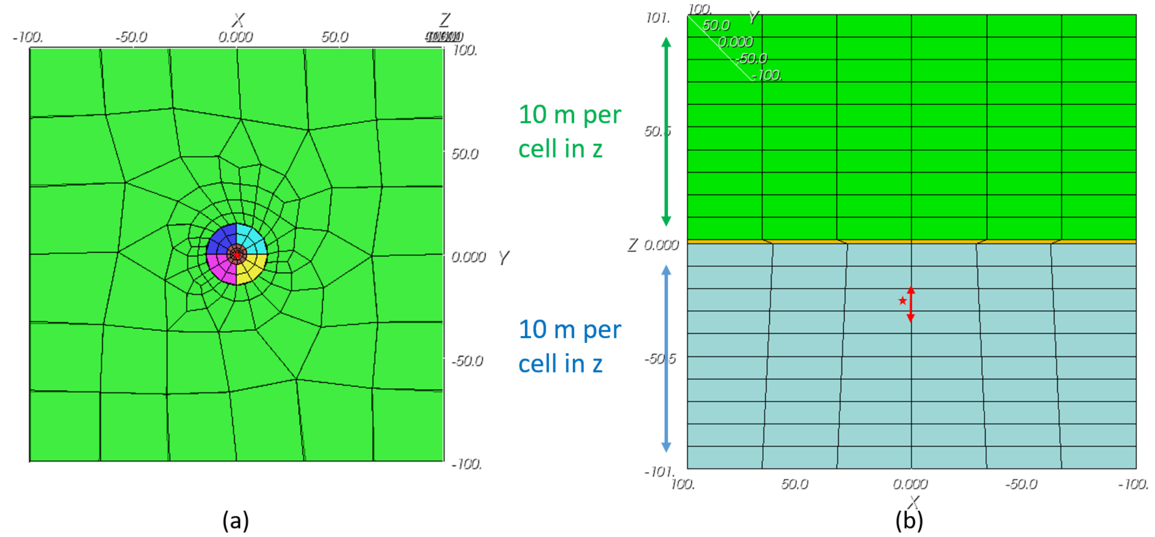


FIGURE 2.12: Nearly homogeneous meshes for the SF solver

2.5 Riemann Transmission Condition

The accuracy and convergence of the DGFD method relate directly to the transmission condition (TC), which is used to ensure the tangential fields continuities across the interfaces as well as supply numerical flux to the subdomain systems to correctly model the boundary effects. The widely used TC in DDM is the first-order Robin transmission condition [Li and Jin (2007); Zhao et al. (2007)], and then is improved to a few high-order TCs [Peng and Lee (2012)]. Another category of the TC is the RTC and origins from the discontinuous Galerkin time domain (DGTD) method with Riemann solver (upwind flux) to model the flux [Luo et al. (2013); Sun et al. (2017)].

A standard way to develop the upwind flux scheme is to use an operator-splitting method [Gonzalez (2013)]. From time domain Maxwell curl equations

$$\mu \frac{\partial \mathbf{H}}{\partial t} + \nabla \times \mathbf{E} = 0 \tag{2.20}$$

$$\epsilon \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{H} + \sigma_e \mathbf{E} = 0 \tag{2.21}$$

we can separate it into two sub-problems by using $\nabla = \frac{\partial}{\partial n}\hat{n} + \nabla_s$,

$$\text{Sub-problem 1:} \quad \mu\frac{\partial \mathbf{H}}{\partial t} + \frac{\partial}{\partial n}\hat{n} \times \mathbf{E} = 0, \epsilon\frac{\partial \mathbf{E}}{\partial t} - \frac{\partial}{\partial n}\hat{n} \times \mathbf{H} = 0 \tag{2.22}$$

$$\text{Sub-problem 2:} \quad \mu\frac{\partial \mathbf{H}}{\partial t} + \nabla_s \times \mathbf{E} = 0, \epsilon\frac{\partial \mathbf{E}}{\partial t} - \nabla_s \times \mathbf{H} + \sigma_e\mathbf{E} = 0 \tag{2.23}$$

The first sub-problem is a homogeneous conservation law, imposing a discontinuous Riemann problem at the interface between two subdomains; while the second sub-problem is a fully continuous problem, which can be abandoned for our derivation of upwind flux. Therefore, the problem degenerates to a 1D problem where the waves are only propagating perpendicular to the interface. Then the two vector equations of sub-problem 1 can be separated to four scalar equations

$$\mu\frac{\partial \mathrm{H}_{t1}}{\partial t} - \frac{\partial \mathrm{E}_{t2}}{\partial n} = 0 \tag{2.24}$$

$$\mu\frac{\partial \mathrm{H}_{t2}}{\partial t} + \frac{\partial \mathrm{E}_{t1}}{\partial n} = 0 \tag{2.25}$$

$$\epsilon\frac{\partial \mathrm{E}_{t1}}{\partial t} + \frac{\partial \mathrm{H}_{t2}}{\partial n} = 0 \tag{2.26}$$

$$\epsilon\frac{\partial \mathrm{E}_{t2}}{\partial t} - \frac{\partial \mathrm{H}_{t1}}{\partial n} = 0 \tag{2.27}$$

$$\tag{2.28}$$

where $t1$ and $t2$ represent two tangential directions of the interface and conform to the left hand rule together with $\hat{n}$.

If make $\boldsymbol{u} = (\mathrm{H}_{t1}, \mathrm{H}_{t2}, \mathrm{E}_{t1}, \mathrm{E}_{t2})^T$, the hyperbolic system can be represented by

$$\frac{\partial \boldsymbol{u}}{\partial t} + \mathbf{A}\frac{\partial \boldsymbol{u}}{\partial n} = 0 \tag{2.29}$$

The eigenvalues and corresponding eigenvectors of matrix $\mathbf{A}$ are

$$\lambda_1 = \lambda_2 = \frac{-1}{\sqrt{\mu\epsilon}}, \quad \lambda_3 = \lambda_4 = \frac{1}{\sqrt{\mu\epsilon}}; \tag{2.30}$$

$$v_1 = (0, -\mathrm{Y}, 1, 0)^T, v_2 = (\mathrm{Y}, 0, 0, 1)^T, v_3 = (0, \mathrm{Y}, 1, 0)^T, v_4 = (-\mathrm{Y}, 0, 0, 1)^T \tag{2.31}$$

where $\mathrm{Y} = \sqrt{\epsilon/\mu}$ is the wave admittance. The eigenvalues amplitude is the wave speed in corresponding material; plus signs represent waves going outside and minus signs represent waves coming back. The materials at different sides of the interface can be identified by matrix $\mathbf{A}^i$ and $\mathbf{A}^j$. Solving the Riemann problem is equivalent to impose the Rankine-Hugoniot jump condition at both sides of the interface; the Rankine-Hugoniot jump condition states that the jumps of the solution at each side of the interface is a linear combination of eigenvectors at that side, which leads to one intermediate state ($\boldsymbol{u}^*$) that constitutes the actual solution of the Riemann problem.

We then define two internal states $\boldsymbol{u}^i$ and $\boldsymbol{u}^j$ inside corresponding domains and express the jumps between intermediate states and internal states as a linear combination of eigenvalues. Note that the direction of jumps should conform to the eigenvectors physical direction (demonstrated by eigenvalues) as follows,

$$\boldsymbol{u}^* - \boldsymbol{u}^i = \alpha_1^i v_1^i + \alpha_2^i v_2^i \tag{2.32}$$

$$\boldsymbol{u}^j - \boldsymbol{u}^* = \alpha_3^j v_3^j + \alpha_4^j v_4^j \tag{2.33}$$

where $\alpha_1^i, \alpha_2^i, \alpha_3^j, \alpha_4^j$ are scalar coefficients. Then we can get the general form of the solution of the Riemann problem as

$$\hat{n} \times \mathbf{E}^* = \frac{\hat{n} \times (\mathrm{Y}^i\mathbf{E}^i + \mathrm{Y}^j\mathbf{E}^j) + \hat{n} \times \hat{n} \times (\mathbf{H}^j - \mathbf{H}^i)}{\mathrm{Y}^i + \mathrm{Y}^j} \tag{2.34}$$

$$\hat{n} \times \mathbf{H}^* = \frac{\hat{n} \times (\mathrm{Z}^i\mathbf{H}^i + \mathrm{Z}^j\mathbf{H}^j) - \hat{n} \times \hat{n} \times (\mathbf{E}^j - \mathbf{E}^i)}{\mathrm{Z}^i + \mathrm{Z}^j} \tag{2.35}$$

To review the RTC, let's revise the weak form of the Helmholtz equation for the whole domain to that for the subdomain i as follows

$$\begin{aligned}\frac{1}{j\omega\mu_0}\int_{\Omega^i}(\nabla\times\mathbf{\Phi})\cdot(\tilde{\mu}_r^{-1}\nabla\times\mathbf{E})dV+j\omega\epsilon_0\int_{\Omega^i}\mathbf{\Phi}\cdot\tilde{\epsilon}_r\mathbf{E}dV \\ -\int_{\partial\Omega\cap\partial\Omega^i}\mathbf{\Phi}\cdot(\hat{n}\times\mathbf{H})dS-\sum_{\Gamma^{ij}\in\partial\Omega^i}\int_{\Gamma^{ij}}\mathbf{\Phi}\cdot(\hat{n}\times\mathbf{H}^*)dS \\ =-\int_{\Omega}\mathbf{\Phi}\cdot\mathbf{J}dV-\frac{1}{j\omega\mu_0}\int_{\Omega}(\nabla\times\mathbf{\Phi})\cdot(\tilde{\mu}_r^{-1}\mathbf{M})dV \quad (2.36)\end{aligned}$$

By comparing with the whole domain governing equation, except the change of the domain of integration, the only difference is the appearance of the boundary integral for the interfaces Γ^{ij}. One may notice that the integral term for domain interfaces has one more star than the integral term for exterior boundaries. The reason is that the exterior boundary fields $\hat{n}\times\mathbf{H}$ are totally determined by the fields inside the computation domain, while the interface fields $\hat{n}\times\mathbf{H}^*$ are determined by both the fields inside subdomains i and j. Therefore, how to evaluate the numerical flux $\hat{n}\times\mathbf{H}^*$ is crucial to the accuracy and convergence of the modeling. Inspired by the upwind flux, (2.47) is employed to calculate the flux. However, the Helmholtz equation itself only enforces the subdomain i and subdomain j share the same numerical flux term, but leaves the electric fields and boundary magnetic fields in different subdomains independent to each other. To enforce the tangential continuity of the fields across the domain interfaces, an explicit TC should be imposed. Most TCs directly equate the linear combinations of the tangential electric and magnetic fields of adjacent subdomains, but here we introduce a different derivation for it. Recall the unused upwind flux equation (2.46), which is an intermediate condition between $\hat{n}\times\mathbf{E}^i$ and $\hat{n}\times\mathbf{E}^j$. If we directly equate boundary tangential $\hat{n}\times\mathbf{E}^i$ and $\hat{n}\times\mathbf{E}^j$ to be $\hat{n}\times\mathbf{E}^*$, we will have

$$\hat{n} \times \mathbf{E}^i = \hat{n} \times \mathbf{E}^* = \frac{\hat{n} \times (\mathrm{Y}^i\mathbf{E}^i + \mathrm{Y}^j\mathbf{E}^j) + \hat{n} \times \hat{n} \times (\mathbf{H}^j - \mathbf{H}^i)}{\mathrm{Y}^i + \mathrm{Y}^j} \tag{2.37}$$

$$\hat{n} \times \mathbf{E}^j = \hat{n} \times \mathbf{E}^* = \frac{\hat{n} \times (\mathrm{Y}^i\mathbf{E}^i + \mathrm{Y}^j\mathbf{E}^j) + \hat{n} \times \hat{n} \times (\mathbf{H}^j - \mathbf{H}^i)}{\mathrm{Y}^i + \mathrm{Y}^j} \tag{2.38}$$

Rearrange the (2.37) and (2.38), we will get

$$\mathrm{Y}^j(\hat{n} \times \mathbf{E}^i) + \hat{n} \times \hat{n} \times \mathbf{H}^i = \mathrm{Y}^j(\hat{n} \times \mathbf{E}^j) + \hat{n} \times \hat{n} \times \mathbf{H}^j \tag{2.39}$$

$$\mathrm{Y}^i(\hat{n} \times \mathbf{E}^j) - \hat{n} \times \hat{n} \times \mathbf{H}^j = \mathrm{Y}^i(\hat{n} \times \mathbf{E}^i) - \hat{n} \times \hat{n} \times \mathbf{H}^i \tag{2.40}$$

which is exactly the same to the first-order TC proposed in [Zhao et al. (2007)]. By solving (2.39) together with (2.40), we can have

$$\hat{n} \times \mathbf{E}^j = \hat{n} \times \mathbf{E}^i \tag{2.41}$$

$$\hat{n} \times \hat{n} \times \mathbf{H}^j = \hat{n} \times \hat{n} \times \mathbf{H}^i \tag{2.42}$$

indicating that the (2.37) and (2.38) actually impose the tangential continuity for both the electric and magnetic fields. Therefore, the RTC employs the general first TC to enforce the tangential field continuity, but models the numerical flux term $\hat{n} \times \mathbf{H}^*$ with a universal Riemann solver instead of roughly make it equal to the local, adjacent or the average of the tangential magnetic fields.

2.6 Domain Decomposition Method

To elaborate the work flow for the DDM, we take a sphere scattering case as example, which resides in a three layer formation. To start with, we consider only the three layer formation first as shown in Fig. 2.13. A electric dipole at 100 kHz polarized in all three directions is put at the origin; an array of receivers is placed at -4 m $\leqslant x \leqslant 4$ m, $y = 1$ m, $z = 0$ m. For the top and bottom layers, $\epsilon_{r1} = \epsilon_{r3} = 40$, $\mu_{r1} = \mu_{r3} = 1$, $\sigma_{r1} = \sigma_{r3} = 0.05$ S/m, and the skin-depth is 7.13 m; for the middle layer, $\epsilon_{r2} = 80$, $\mu_{r2} = 1$, $\sigma_{r2} = 0.5$ S/m, and the skin-depth is 2.25 m. With the help

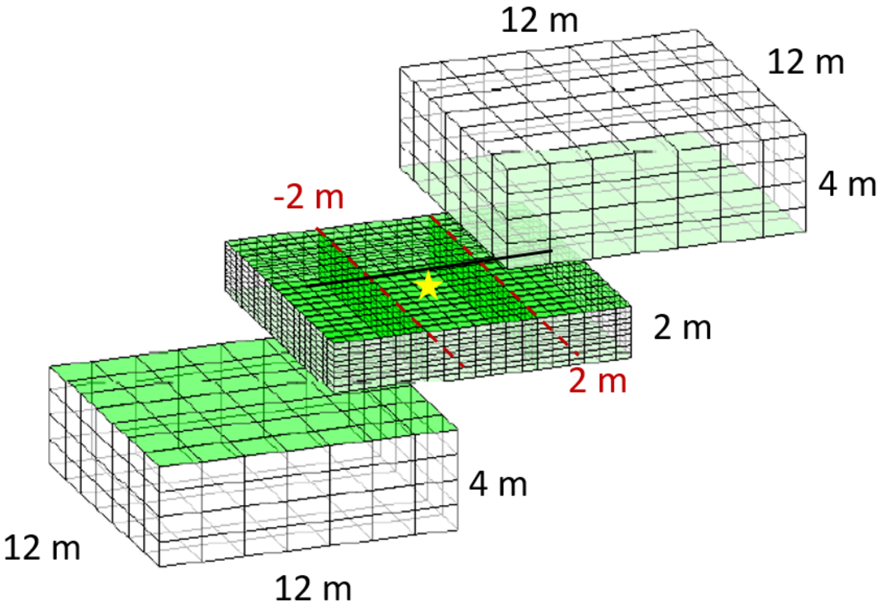


FIGURE 2.13: Diagram of the three layer formation

of the analytical solution for homogeneous background of the middle layer's property, the relative error can be safely kept less than 40 -dB if put the PEC boundary at $x = \pm 8$ m and $y = \pm 8$ m. To further squeeze the computational domain, a second-order scattering boundary condition (SBC) is employed [Peterson (1988)]

$$\hat{n} \times (\nabla \times \mathbf{E}) - jk\hat{n} \times (\mathbf{E} \times \hat{n}) - \frac{1}{2jk}\nabla \times (\hat{n}\hat{n} \cdot (\nabla \times \mathbf{E})) = 0 \qquad (2.43)$$

Therefore, the SBC is put at $x = \pm 6$ m, $y = \pm 6$ m and $z = \pm 5$ m. A comparison between SBC and PEC for the same computational domain will be provided later. Currently, because no object is present, structured meshes can be applied to keep the sparsity of system matrix for high-order basis functions, as well as to speed up the matrix assembling process. Further, since the middle layer has smaller skin-depth and contains both a source and receivers, it is appealing to nonconformally decompose the domain at the formation layer interfaces to allow the middle layer discretized by denser meshes. The mesh information is also shown in Fig. 2.13, with the mesh in the top and bottom layers (the middle layer) 1.7 m (0.63 m) in x,y directions and 1 m (0.22 m) in z direction. By employing the third-order basis function for the top and bottom layers and the second-order basis function for the middle layer, the DoFs are about 19,000 for top and bottom layers and 90,000 for the middle layer. To balance the DoFs in different subdomains, we decompose the middle layer again but conformally this time at $x = \pm 2$ m to make each of them about 30,000 DoFs. One may notice the middle layer employ low-order but dense meshes to avoid the source singularity, which can be replaced by more efficient high-order but coarse meshes if SF solver is employed for the middle layer. For the top and bottom layers, however, since they don't include sources and have different formation than the middle layer, the total field (TF) solver is more suitable. Therefore, we propose a TF/SF mixed solver [Sun et al. (2018)] to solve different subdomains with different solvers.

2.7 Total Field/Scattered Field Mixed Solver

It is not hard to show that the mixed solver will only change the RTC by adding or subtracting a incident field term for adjacent TF-based subdomains and SF-based subdomains [Sun et al. (2018)]. For a TF-based subdomain i coupled with a SF-based subdomain j, the RTC will be revised as

$$\hat{n} \times \mathbf{E}^* = \frac{\hat{n} \times (\mathrm{Y}^i \mathbf{E}^i + \mathrm{Y}^j (\mathbf{E}_s^j + \mathbf{E}_b)) + \hat{n} \times \hat{n} \times (\mathbf{H}_s^j + \mathbf{H}_b - \mathbf{H}^i)}{\mathrm{Y}^i + \mathrm{Y}^j} \tag{2.44}$$

$$\hat{n} \times \mathbf{H}^* = \frac{\hat{n} \times (\mathrm{Z}^i \mathbf{H}^i + \mathrm{Z}^j (\mathbf{H}_s^j + \mathbf{H}_b)) - \hat{n} \times \hat{n} \times (\mathbf{E}_s^j + \mathbf{E}_b - \mathbf{E}^i)}{\mathrm{Z}^i + \mathrm{Z}^j} \tag{2.45}$$

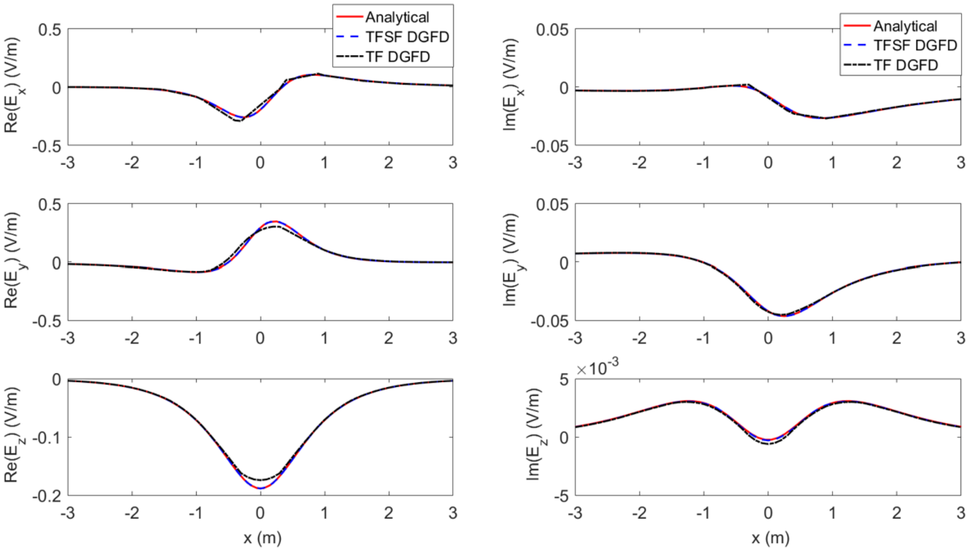


FIGURE 2.14: Comparison between the TF solver and the TF/SF mixed solver for the dense mesh setup

For a SF-based subdomain i coupled with a TF-based subdomain j, the RTC will be revised as

$$\hat{n} \times \mathbf{E}_s^* = \frac{\hat{n} \times (\mathrm{Y}^i \mathbf{E}_s^i + \mathrm{Y}^j (\mathbf{E}^j - \mathbf{E}_b)) + \hat{n} \times \hat{n} \times (\mathbf{H}^j - \mathbf{H}_b - \mathbf{H}_s^i)}{\mathrm{Y}^i + \mathrm{Y}^j} \tag{2.46}$$

$$\hat{n} \times \mathbf{H}_s^* = \frac{\hat{n} \times (\mathrm{Z}^i \mathbf{H}_s^i + \mathrm{Z}^j (\mathbf{H}^j - \mathbf{H}_b)) - \hat{n} \times \hat{n} \times (\mathbf{E}^j - \mathbf{E}_b - \mathbf{E}_s^i)}{\mathrm{Z}^i + \mathrm{Z}^j} \tag{2.47}$$

To validate the TF/SF mixed solver, the same mesh setup as shown in Fig. 2.13 is used for both the TF and TF/SF solvers; the corresponding electric field distributions are compared with the analytical solution of the three layer formation in Fig. 2.14, with the relative error 11.7% for the TF solver and 0.37% for the TF/SF solver. It is clear that the TF results for the three central meshes differ from the analytical result and can be attributed to the source singularity.

Further, by using the TF/SF solver for the mesh in Fig. 2.14, we can compare the performance between the SBC and the PEC as shown in Fig. 2.15. The relative error for the SBC is 0.37% while for the PEC is 1.61%; the shift of the imaginary part

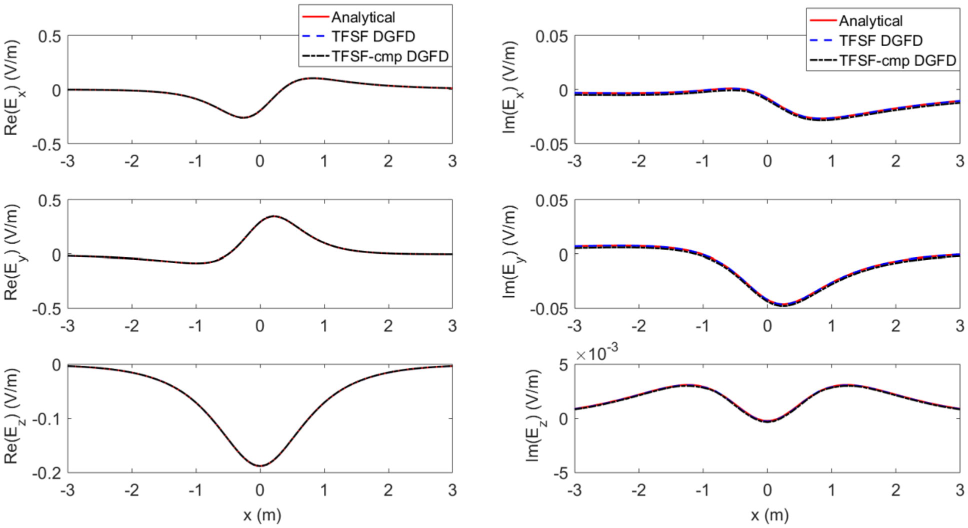


FIGURE 2.15: Comparison between the SBC and the PEC for the TF/SF mixed solver of the dense mesh setup

of the electric field from the PEC is visible and can be attributed to the inadequate decay distance between the receivers and the PEC boundary.

To demonstrate the singularity removal ability of the TF/SF mixed solver, we change the meshes in the middle layer to be the same as other two layers as shown in Fig. 2.16 with the fourth-order basis function. The DoF for the middle layer then becomes about 25,000. With the new mesh setup, the TF and TF/SF solvers are employed again and their corresponding electric field results are compared in Fig. 2.17. Clearly, the TF solver totally fails in the near source region, while the TF/SF can still maintain a 0.3% relative error. When employing the BiCGStab(L=10) iteration method to solve the TF/SF solver generated interface matrix for a 10^{-4} convergence tolerance, each iteration only takes 10 s for the coarser meshes while takes 25 s for the denser meshes, which is nearly linear proportional to the total DoFs.

With the three layer formation frames, we then place a sphere object of r_s = 0.4 m, ϵ_{rs} = 80, σ_s = 0.001 S/m at (1,0,0.2) m inside the middle layer. A new subdomain is decomposed just wrapping the sphere as shown in Fig. 2.18, where

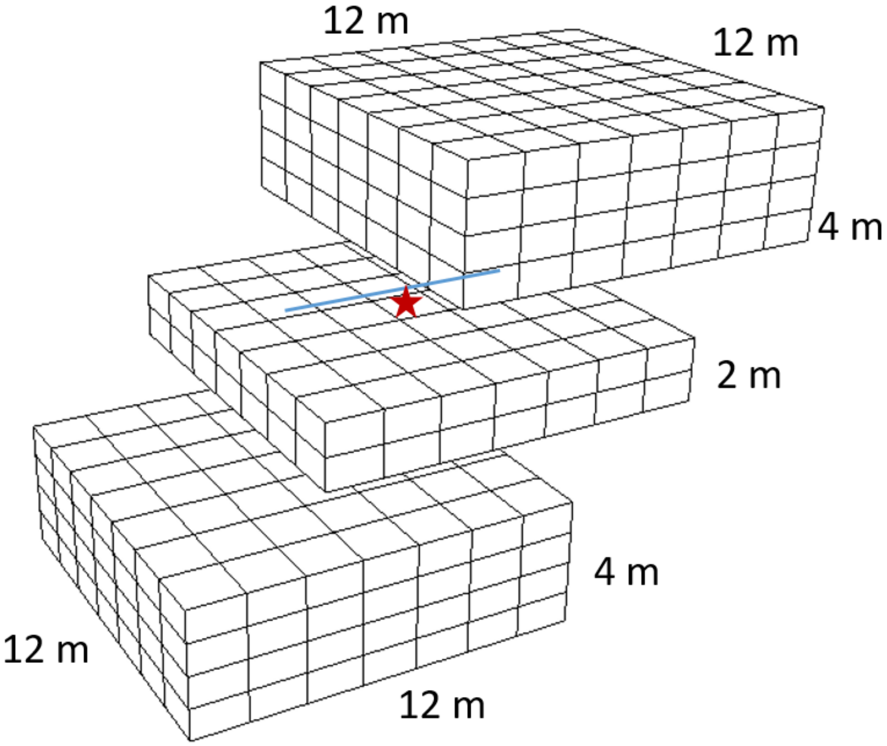


Figure 2.16: Diagram of the three layer formation with larger elements in the middle layer

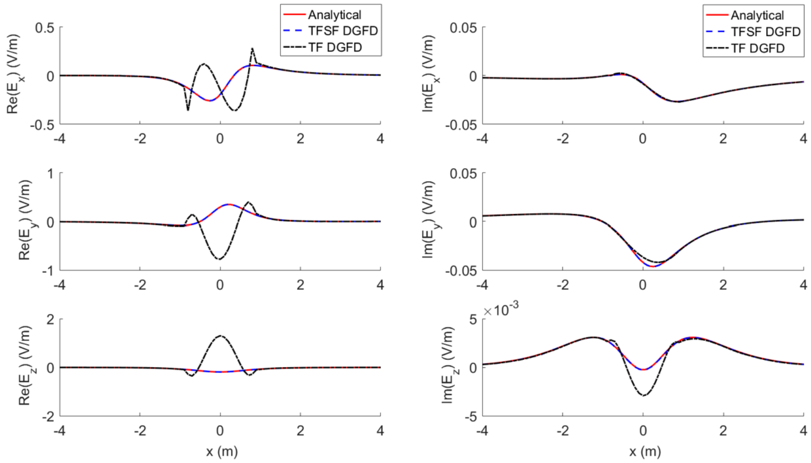


Figure 2.17: Comparison between the TF and TF/SF solvers for the coarse mesh setup

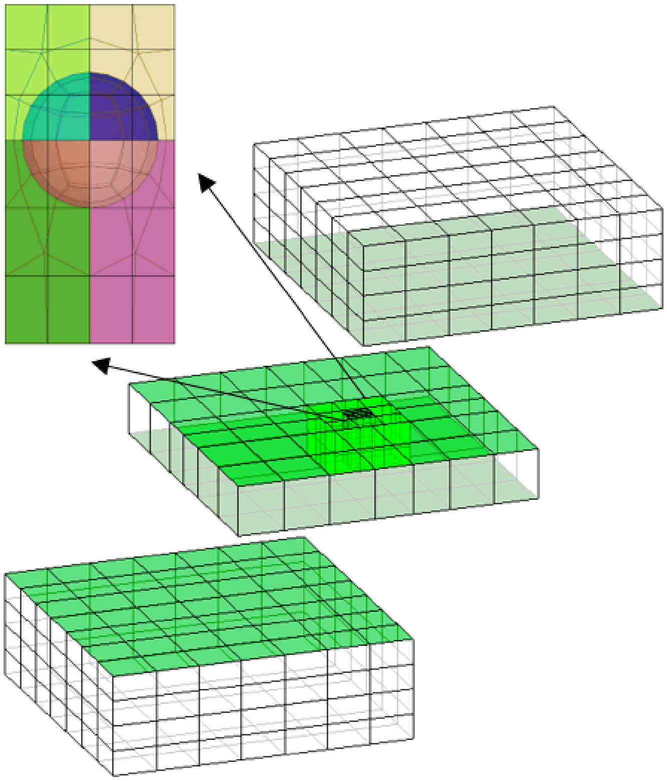


Figure 2.18: Diagram of the three layer formation wrapping a spherical object in the middle layer

the sphere subdomain employs the second-order basis function and second-order geometry mapping to capture the curved surface, the transition shell domain employs the sixth-order basis function to capture strong near object scattered fields, and the middle (top/bottom) layers employ the fourth(third)-order basis functions. The DoFs of the middle layer from inside out are about 9,000, 16,000 and 12,000 and other two layers are about 19,000. The middle transition layer and middle outer layer employ the SF solver to relieve the source singularity and other regions use the total field to avoid the equivalent source assembling. Each iteration will take about 18 s and only 2 steps are required for a 10^{-4} convergence tolerance and a 1%

relative error compared to the analytical solution for the incident fields, while the COMSOL needs more than 700,000 tetrahedrons to achieve the same accuracy for the incident fields and runs about one and half hour. The SF solver here relieves the mesh burden for the source, and only geometries need to be captured by dense unstructured meshes; therefore, the resultant system matrix can be smaller than normal FEM due to the spectral convergence of the SEM and the orthogonality of brick meshes.

The calculated scattered electric and magnetic fields from the sphere are compared with the COMSOL multiphysics are plotted in Fig. 2.19 and Fig. 2.20. Both the DGFD and COMSOL require two computation; one for the incident fields and one for the total fields. The relative error are about 5% for both fields. The enlarged relative errors can be attributed to the small amplitude of the scattered fields, which serve as the denominators for the relative errors. Notice that the COMSOL needs more than 700,000 tetrahedrons to achieve less than 1% relative error compared to the analytical solution for the incident fields and runs about one and half hour.

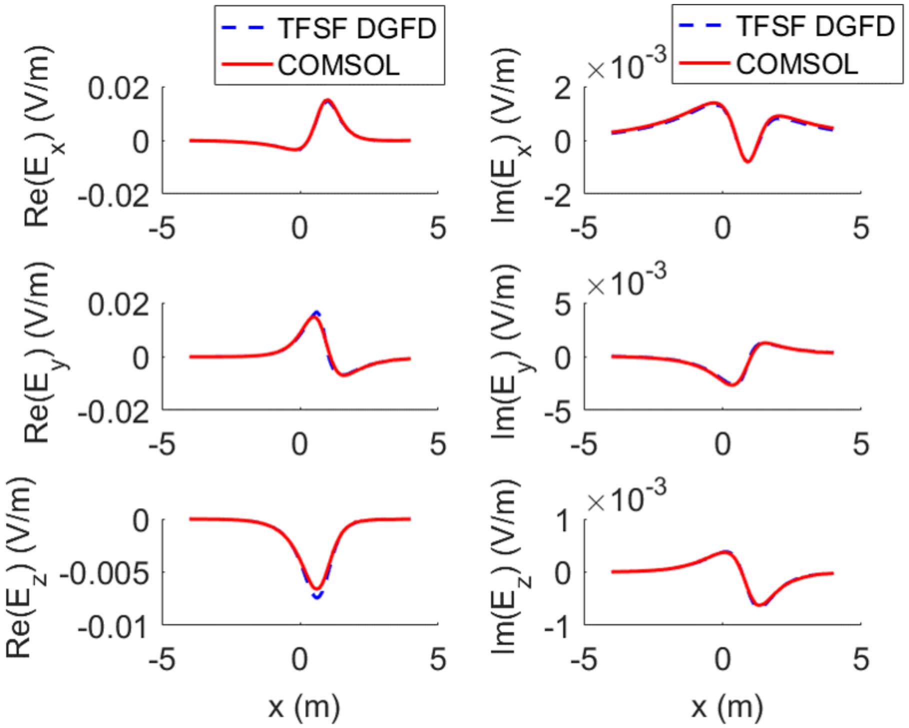


FIGURE 2.19: Comparison between the TF/SF equipped DGFD and COMSOL for the electric field

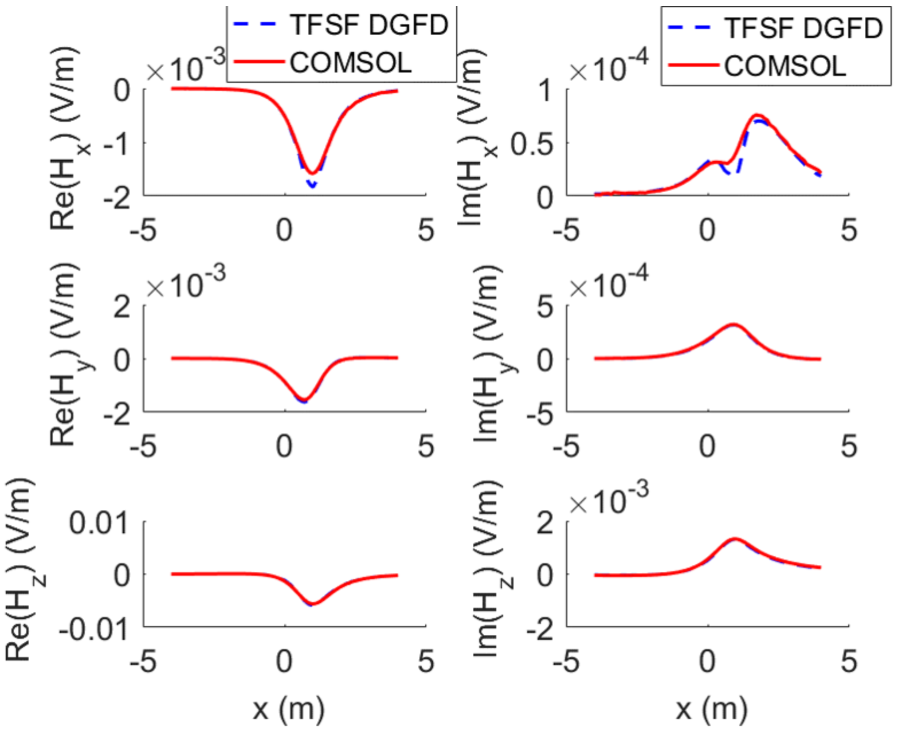


Figure 2.20: Comparison between the TF/SF equipped DGFD and COMSOL for the magnetic field

3

DGFD Method for Well Logging Problems

Part of this chapter was originally published in [Zhang, R., Sun, Q., Wu, Z., Fang, Y., Hu, Y., Huang, W.-F., and Liu, Q. H. (2019), Fast Induction Logging Modeling With Hierarchical Sudoku Meshes Based on DGFD, *IEEE Geoscience and Remote Sensing Letters*, 16, 1683–1687] and [Zhang, R., Wu, Z., Sun, Q., Zhuang, M., Cai, Q.-M., Wang, D., and Liu, Q. H. (2019), Memory-efficient 3-D LWD solver with the flipped total field/scattered field-based DGFD method, *IEEE Geoscience and Remote Sensing Letters*], and is reprinted with permission. I designed the algorithms, developed the codes and wrote the articles.

Well logging is an essential technique and widely used in oil and gas exploration and development. Simulation of these measurements in complex subsurface formation and borehole environments usually requires 3D modeling techniques, which are very memory demanding and time consuming. In this chapter, I am going to introduce two improvements towards the forward modeling of the well logging problems based on the DGFD method, both of which can reduce the memory consumption as well as computation time.

3.1 Hierarchical Sukudo Mesh for Vertical Well Logging

This section extends the DGFD-based DDM to model the vertical well environment accurately with a newly developed hierarchical sudoku mesh. Both the borehole and mud invasion effects can be modeled by this framework efficiently. Furthermore, deep reading measurements can be modeled conveniently by inserting arbitrarily shaped objects into the hierarchical sudoku meshes; the capability to distinguish a cavity saturated with either oil or water with different borehole-object distances are then studied for a deep reading tool in an open borehole carbonate environment. The DGFD with hierarchical sudoku meshes shows six times faster speed than the traditional FEM for the deep reading case.

3.1.1 Background

Induction logging is an essential approach in oil and gas industry for the formation evaluation, where numerical modeling can help to design induction tools and interpret data. The 1.5D and 2.5D simulation methods are well developed and widely used in the industry; however, with the ever large depth of investigation (DOI) of the deep reading induction tools, reservoir-scale mapping [Hu et al. (2017); Seydoux et al. (2014)] and hydraulic fracture mapping [Fang et al. (2018); Pardo and Torres-Verdín (2013)] become possible, which generally cannot be simplified to lower dimensional problems.

To numerically model the objects' EM responses under deep reading induction tools in an open borehole in 3D, however, is very challenging. First, the borehole is a very narrow cylindrical hole (5∼25 centimeter radius) while the DOI can range from a few meters to up to 30 meters, raising a multiscale problem in transverse direction. Second, to guarantee enough signal responses for the large DOI, the transmitter-to-receiver (TR) spacing of the logging tool should also range from meters to thirty meters long, which sets a lower bound of the simulated borehole length for a fixed tool position. To compute the logging curve, however, the tool needs to be moved along with the borehole, which requires us to either remodel the problem for every tool position [Zeng et al. (2018b)], or include the tool trajectory to our modeled region [Fang et al. (2018)]; the first strategy breaks the logging curve problem into multiple independent fixed-tool problems without sharing any intermediate stage; the second strategy processes and assembles matrices for the whole model, and then solves the same linear system for different tool positions. The drawback of the first strategy is the reduplicate mesh processing and matrix assembling, and correspondingly high computational cost per logging point. On the contrary, the second strategy shares the mesh processing and matrix assembling, and therefore lowers computational cost per logging point, under the condition that the larger linear system can still be solved efficiently.

To leverage the second strategy, many efforts are put to develop fast solvers for the large linear system. The VIE-based method is traditionally used to minimize the unknowns of the linear system, which only discretizes the region different from the Green's function modeled background [Fang et al. (2018); Yang et al. (2015)]. The Green's function leads to high efficiency and high accuracy, but restricts the application to homogeneous or layered media. For induction tool modeling in an open borehole under complicated geophysical environments, the FEM is more universal at the cost of modeling the anomaly, the borehole, together with the formation. To alleviate the large sparse matrix generated from the FEM [Ma et al. (2014b); Pardo et al. (2007)], the DGFD-based DDM is introduced in [Sun et al. (2017)] to split the computational domain into multiple subdomains, thus greatly reducing the computational complexity. However, the case study of borehole environments in [Sun et al. (2017)] is very preliminary, not showing the efficiency of the DDM and also not scalable for deep reading tool modeling. To fill this gap, we develop a sudoku mesh

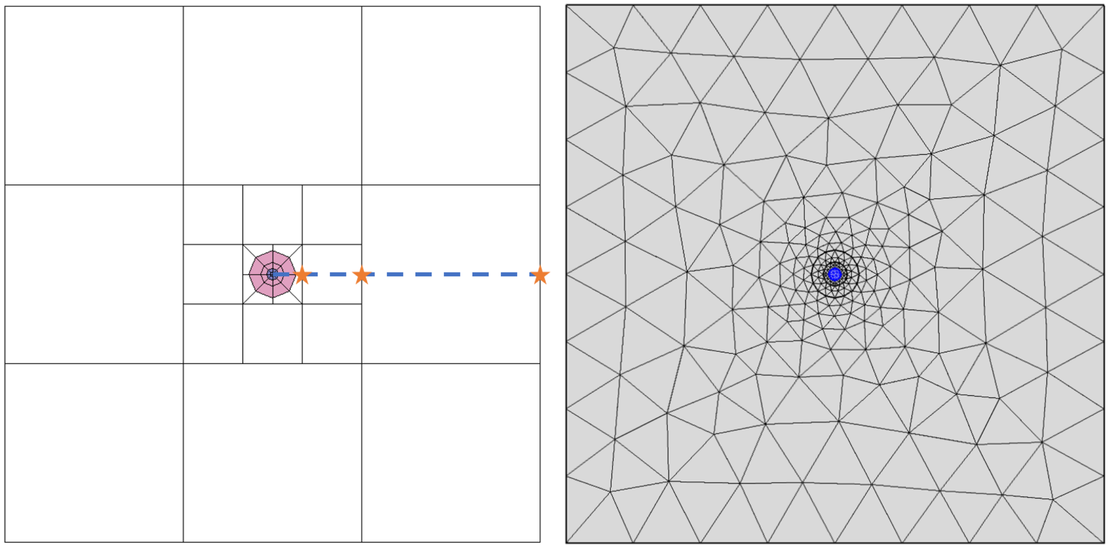


Figure 3.1: Comparison of the top-views between the hierarchical sudoku mesh (left) and the tetrahedron mesh from COMSOL (right)

to extend the DGFD-based DDM to model the open borehole environment with high efficiency.

3.1.2 Hierarchical Sudoku Mesh

The formulas for the DGFD-based DDM are provided in (2.36). To capture the geometrical and wave property of the open borehole environments effectively and efficiently, instead, a non-conformal gradient structured mesh design concept, the hierarchical sudoku mesh, is introduced here, which can generate a well-behaved DDM preconditioned interface system with fewer DoFs than traditional FEM-based linear system.

Because the borehole can usually be modeled as a long straight cylinder, we can focus on its transverse cross section plane first and design a 2D mesh; by extruding the 2D mesh along with the longitudinal direction, we will then obtain a 3D mesh. Let's start from the transverse cross section of a practical borehole with 20 cm radius; quadrilaterals with the second-order geometry mapping are employed to model the cross section of borehole (blue) and invasion zone (red) as the left figure of Fig. 3.1. Traditionally, unstructured tetrahedra are employed to model the borehole resistivity measurement because the mesh size can progressively increase outwards from the borehole [Um et al. (2015)]. To model deep reading tools, however, larger formation and anomaly regions need to be included, where high-order basis functions are usually favored over their low-order counterparts due to the high-order convergence rate and reduced DoFs. One consequence of the high-order basis function is the enlarged non-zero blocks in the global matrix, because generally all basis functions of the same direction within one element are coupled together. To reduce the non-zero entries in the global matrix, the spectral element method (SEM) employs hexahedra in seismic modeling [Komatitsch and Tromp (1999)] with Lagrange interpolants of nodal points at the Gauss-Legendre-Lobatto (GLL) quadrature, and the GLL integral can then guarantee the mass matrices diagonal, even for well-defined unstructured hexhedra, because the mapping of scalar basis functions does not involve Jacobian matrix multiplication or division. In computational EM, the orthogonality of the GLL polynomials [Lee et al. (2006)] with unstructured hexahedra degrades because the covariant (contravariant) mapping of the curl-conforming basis functions (the curl of curl-conforming basis functions) makes the elemental mass (stiffness) matrix full. However, structured meshes can still keep their sparsity to the maximum extent; therefore, the resultant global matrix will have fewer non-zero entries than those generated by unstructured meshes or other basis functions given the same mesh size and order, reducing the memory overhead. To make full use of the structured hexahedra with high-order GLL polynomials, a quadrilateral buffer layer of the second-order geometry mapping is employed to transition the internal unstructured hexahedra to the external structured hexahedra.

To realize progressively increasing structured meshes, a DDM-based hierarchical sudoku mesh is introduced here. As we know, the sudoku includes nine equal squares; at the beginning, the transition layer wrapped borehole/invasion, which can be thought as the zeroth-level sudoku, locates at center, surrounded by the first-level sudoku, a ring of eight squares; next, the first-level sudoku serves as the center square, surrounded by the second-level sudoku, a ring of eight squares again; by following the same spirit, the sudoku structure grows outwards, with a linearly growing mesh number and an exponential growing computational domain, both in terms of the number of layers. To mathematically demonstrate the advantage of the hierarchical sudoku mesh, we consider a homogeneous case and focus on the wave amplitude at the intersections (orange stars) of an arbitrary radial line (blue dashed line) and boundaries of the hierarchical sudoku's as shown in Fig. 3.1. For kilohertz induction measurements, the $1/r^3$ term of the spherical waves dominates the attenuation; the amplitude ratios between the adjacent intersections is then approximately a constant $1/3^3$, which conforms to the constant DoFs for the one element span if equipped with same order basis functions.

To extrude the 2D mesh into 3D meshes, the longitudinal span can be different; for example, to extrude all squares into cubes, the span will have a growth rate of 3 for the cubes from inside to outside, and the total number of the structured mesh number will then be bounded by one and a half times the structured mesh number of the first-level sudoku; in other words, the DoFs for the first-level sudoku will dominate over the DoFs for all other high level sudokus, reducing the FEM's burden of modeling background while still capturing the wave diffusion phenomenon. Furthermore, recall that the attenuation coefficient is about $1/3^3$ across one layer of the sudoku, which indicates that if we wrap the computational domain with one more PEC truncated sudoku, the reflection coefficient will be less than $1/3^6$; the reflection is around 0.1% with only $N^{(1)}/3^{(m-1)}$ additional DoFs, where m is the sudoku level and $N^{(1)}$ is the DoFs of the first-level sudoku. Notice that in borehole resistivity measurement, the source usually moves along with the borehole; therefore, to preprocess, assemble and factorize the matrix only once, the meshes of sudokus at all levels should keep invariant along the longitudinal direction and the direct solver can then be applied. However, the direct solver will encounter the memory overflow for large linear systems; in this section, we factorize smaller subdomain matrices once instead of the global matrix and then use them to precondition and solve the global system iteratively. To further reduce the unknowns for the borehole resistivity measurement in the future, sudoku meshes with variant sizes along the borehole direction are promising; two issues, however, need to be addressed: 1. the repeated matrix assembling needs to be accelerated, potentially with the hp-grid-refinement [Pardo et al. (2007)]; 2. a good preconditioner without subdomain matrix factorization is required to reduce the condition number while avoiding repetitive factorization.

To model arbitrarily shaped anomalies, we first locate the sudoku meshes they occupy; then we aggregate the occupied sudoku meshes into rectangular boxes, one box per sudoku level; finally, the rectangular boxes are discretized with unstructured hexahedra to capture the geometry of anomalies, where the meshing density decreases with the increase of the corresponding sudoku level. Although the hexahedra are not as flexible as tetrahedra when modeling complicated geometries, they can model most structures through mapping and sweeping techniques after appropriate geometry decomposition. Further, the non-conformal DDM between the anomaly subdomains and surrounding hierarchical sudoku subdomains allows anomaly subdomains meshed nearly independently, further alleviating the meshing burden for the unstructured hexahedra. The only requirement is that the anomaly subdomain should keep the number of points per wavelenth (PPWs) higher than the same level sudoku region to guarantee the accuracy of the anomaly modeling.

3.1.3 Results and Discussions

To verify the accuracy and efficiency of the hierarchical sudoku mesh for the borehole environment, two kinds of numerical experiments are conducted in this section. We first apply our hierarchical sudoku meshes to a classical layered formation evaluation problem with invasion zones; then, we study the capability of distinguishing a water saturated cavity from an oil saturated cavity by using deep reading tools in carbonate environment. At the linear equation solution phase, a two-step matrix solution procedure [Peng and Lee (2012)] is used to reduce a large linear system involving all the unknowns to an interface unknown system involving only the interface unknowns. To avoid the matrix-matrix division, we do not assemble the interface matrix explicitly, but use a Krylov subspace iterative method, the BiCGStab(L) [Sleijpen and Fokkema (1993)], to perform only matrix-vector multiplications for coupled matrices and forward (backward) substitutions for subdomain L (U) matrices.

Verification: Layered Formation Case

A 15-cm radius borehole with water-based mud of conductivity 5 S/m is modeled in this section with a three-layer formation including the invasion effects. The corresponding electromagnetic and geometric parameters are listed in Table 3.1. Notice that the thickness of the middle layer is 4.5 m while other two layers are half infinite modeled by a 13.5 m-thick layer. To measure the formation resistivity, induced currents need to escape from the invasion zones and inject into the virgin zones, which can be achieved by an triaxial induction tool of 1-m TR spacing at 13 kHz. The lateral boundaries can then be set 10 m away from the borehole with scattering boundary condition to achieve a boundary reflection below 40 dB.

The hierarchical sudoku mesh in Fig. 3.1 (a) is designed as follows: the zeroth-level sudoku models the borehole together with invasion zones; the first-level sudoku wraps the zeroth-level sudoku, modeling the virgin zones; the second-level sudoku

Table 3.1: Parameters for the layered formation model

Layer **[thickness (m)]**	**Formation** **conductivity (S/m)**	**Invasion zone conductivity** **(S/m) [radius (cm)]**
Top [13.5]	0.25	0.8 [30]
Middle [4.5]	0.05	0.4 [61]
Bottom [13.5]	0.5	1.0 [30]

Table 3.2: Model statistics for the layered formation model

Subdomain	**Mesh size [order]**	**Total [Interface] DoFs**
Zeroth-level	$(0.05 \sim 0.3)^2 \times 0.25$ [2^{nd}]	115,752 [15,392]
First-level	1.5^3 [5^{th}]	80,320 [30,560]
Second-level	4.5^3 [5^{th}]	25,660 [2,840]

wraps the first-level sudoku and decays the waves until the boundary. By comparing the sudoku mesh with the traditional tetrahedron mesh as shown in Fig. 3.1 (b), we can find that they share some similarities: 1. Both meshes become coarser away from the borehole, corresponding to the fast attenuation of fields in near-field region; 2. The smallest meshes appear at the center with about 0.05 m side length, which can relieve the singularity caused by the imposed magnetic dipole. However, the differences are also obvious: 1. The sudoku mesh outside the borehole/invasion region is structured, which better supports the high-order GLL basis function due to the orthogonality; therefore the mesh can be much coarser compared to the lower-order tetrahedron mesh; 2. The unstructured mesh inside the zeroth-level sudoku is obtained by sweeping the 2D quadrilaterals along with the longitudinal direction, which not only keeps the orthogonality between the longitudinal vector basis functions and the transverse vector basis functions, but also fits the sweepable borehole/invasion structure naturally; 3. The gradient sudoku mesh involves non-conformal mesh interfaces, requiring appropriate trasnsition condition to take care, while the gradient tetrahedron mesh is conformally connected.

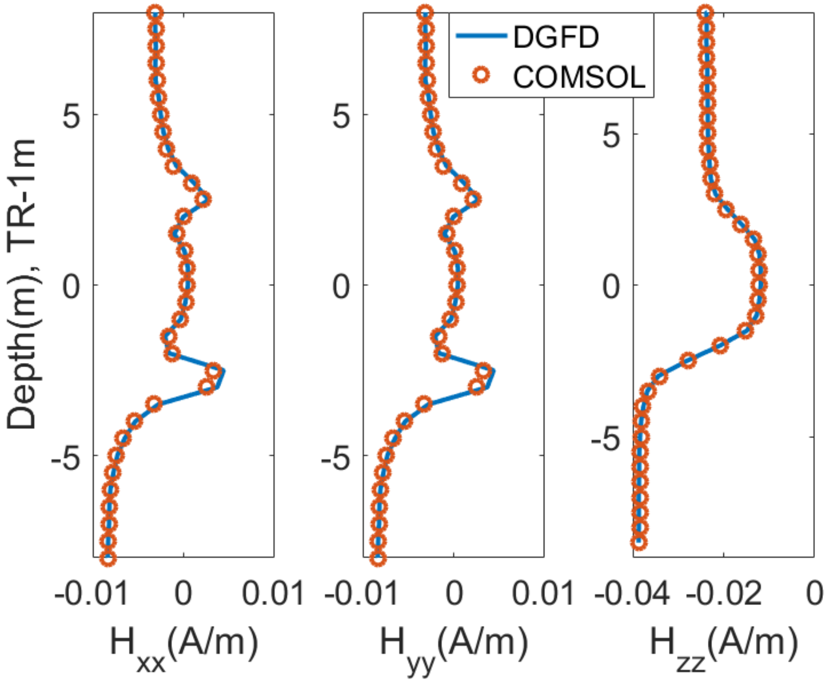


FIGURE 3.2: Comparison of imaginary parts of co-planar components between the DGFD and COMSOL

The corresponding numerical information obtained for the sudoku mesh is provided in Table.3.2. Since the DDM is used as a preconditioner in our DGFD method, the unknowns involved in our iterative solver are only at the interface; therefore, although the total DoFs is above 200,000, the dimension of the actual linear system is below 50,000. On the contrary, COMSOL will need to solve all unknowns together in the linear solution phase, which includes more than 1,000,000 unknowns to achieve the same accuracy.

The co-planar components $\mathrm{H_{xx}}, \mathrm{H_{yy}}, \mathrm{H_{zz}}$ are plotted in Fig. 3.2 to verify the proposed sudoku mesh; note that only the imaginary parts of the fields are plotted, because the real parts are directly coupled with the magnetic dipole and cannot reflect the induction current information; besides, though all nine components are computed, only non-zero co-planar components are shown here. All three components in general agree very well between the sudoku mesh-based DGFD and the

Table 3.3: Model statistics for the deep reading case

Subdomain	**Mesh size [order]**	**Total [Interface] DoFs**
Zeroth-level	$(0.1 \sim 0.5)^2 \times 0.25$ [2^{nd}]	437,152 [96,800]
First-level	1^3 [4^{th}]	435,808 [193,664]
Second-level	3^3 [4^{th}]	145,504 [64,640]
Third-level	9^3 [4^{th}]	46,740 [21,248]
Fourth-level	27^3 [4^{th}]	16,388 [7,424]
Fifth-level	$81^2 \times (54 \sim 81)$ [4^{th}]	6,064 [800]
Anomaly (inner)	$(1 \sim 2)^3$ [2^{nd}]	79,606 [20,960]
Anomaly (outer)	$(2 \sim 4)^3$ [2^{nd}]	32,414 [9,600]

tetrahedron mesh-based COMSOL; a little error occurs for the transverse components at the formation interface, which can be attributed to the accuracy loss of the DDM. When both codes are run on a 8-core server, the 111 independent iteration processes (37 logging points × 3 polarization directions of the source) take about 15 min for DGFD and 50 min for COMSOL.

Application: Cavity Identification by Deep Reading Tools

In hydrocarbon exploration field, the seismic surveys can provide the porosity of formation; for high porosity region, the logging tools are employed to identify whether they are saturated with oil or water. By following the same spirit, we migrate the techniques from formation evaluation to deep investigation, where the seismic surveys can find the cavities deep embedded in the carbonate rocks by porosity, and then the induction tools can distinguish the oil from water by conductivity contrast.

To set up the model, we suppose a composite cavity with the nearest edge 6 m away from the borehole along the positive x axis as shown in Fig. 3.3, where meshes of different sudoku levels are shifted a bit along the z direction to make the details clearer. The cavity includes a frustum of 10-m bottom radius, 5-m top radius and 5-m thickness with a 5-m radius hemisphere on its top. Further, since

Table 3.4: Parameters for the deep reading case

Material	**Conductivity (S/m)**
Carbonate rock	0.001
Oil-based mud borehole	0.05
Water saturated cavity	1
Oil saturated cavity	0.05

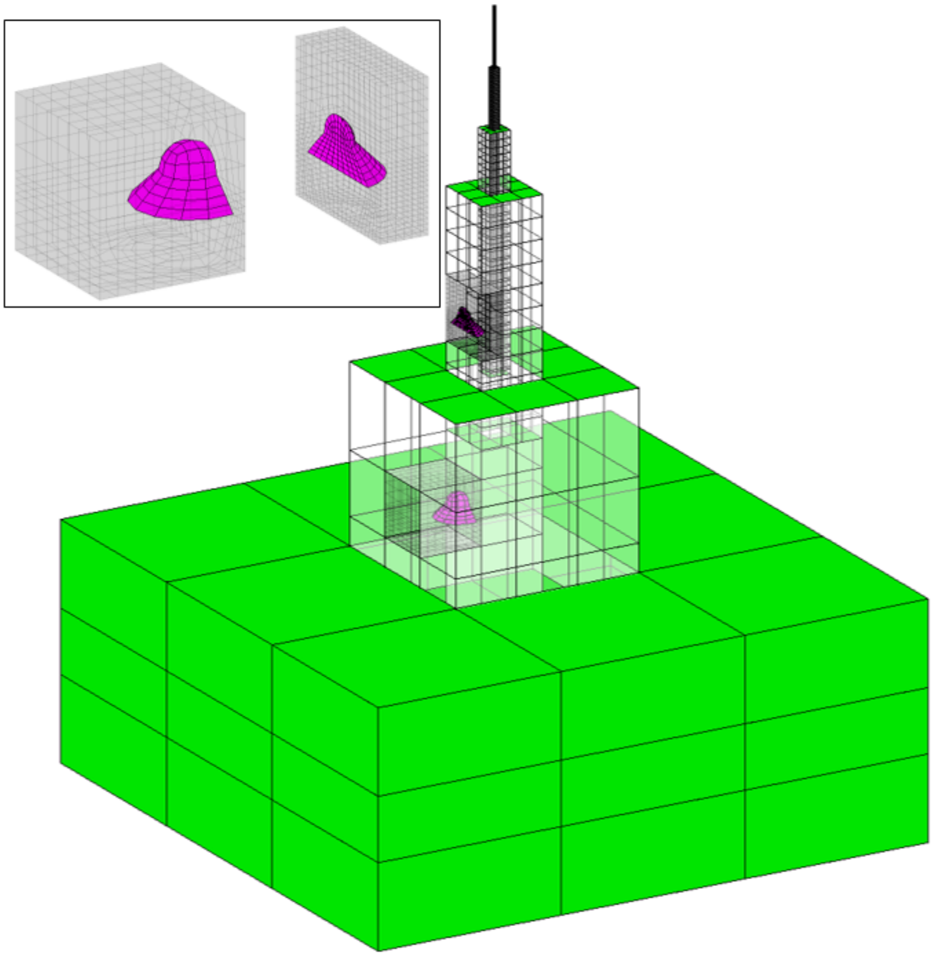


FIGURE 3.3: Hierarchical sudoku mesh of the deep reading case. The cavity (pink regions) is decomposed into two subdomains with different mesh density conforming to the level of adjacent hierarchical sudoku meshes.

the cavity crosses two levels of hierarchical sudoku meshes, it is decomposed into two subdomains with the inner subdomain meshed denser than the other one as shown in the inset. All three dimensions are around 200 m to reduce the boundary effect together with the scattering boundary condition. Detailed mesh grid information in Table 3.3 shows the total DoFs with hierarchical sudoku meshes are 1.20×10^6; to secure the same accuracy, COMSOL employs the second-order basis functions and needs about 12.0×10^6 DoFs. The one order DoF reduction can be attributed to the lower PPWs requirement of the high-order basis functions over the low-order basis functions and the lower PPWs requirement of swept hexahedra over tetrahedra for cylindrical structures under the same order basis functions. Moreover, since the number of interface unknowns reduces by two thirds per sudoku level starting from the first-level sudoku, only about 0.415×10^6 interface unknowns are actually solved by the linear system, which helps to improve the convergence behaviour.

Detailed electromagnetic properties are listed in Table 3.4. To read the deep buried anomaly, the TR spacing of the deep reading tool should be proportional to the required DOI and then larger than that of the traditional induction tool. In this case, we employ a 8-m TR spacing and the dominant triaxial results of 24 logging points are plotted in Fig. 3.4. The DGFD results agree with the COMSOL results well, and only take 3 iterations and 1.5 min per component per logging point to arrive at a relative residue around 3×10^{-6}, while COMSOL needs 8.4 min per component per logging point on a 8-core server.

Let's dive deeper into Fig. 3.4; all the five components have tremendous difference between the responses of an oil saturated cavity and a water saturated cavity, where the water saturated cavity induces larger response than the other cavity due to the larger conductivity contrast. Among co-planar components, H_{xx} has largest normalized response around 50 % (4 %) for the water (oil) saturated cavity. The normalized response of cross-coupled components H_{zx} and H_{xz} are 57 % (8 %) and 50 % (5 %) for the water (oil) saturated cavity, respectively. When focusing on the water induced responses, we can divide the five figures into two groups according to the

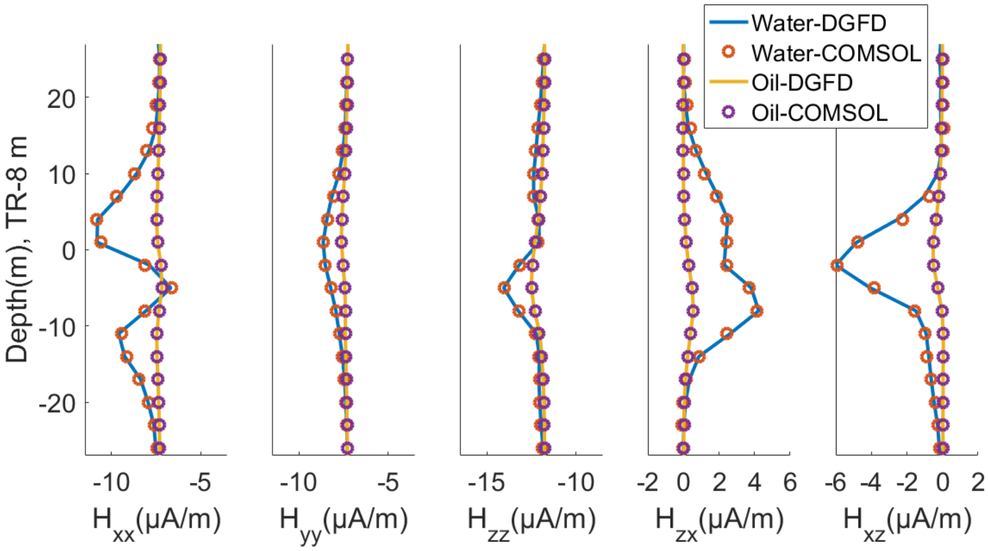


FIGURE 3.4: Dominant components of 8-m TR spacing are plotted to determine a cavity (6 m away from the borehole) saturated with water or oil.

number of peaks. H_{yy} and H_{xz} have one peak, showing that one cavity exists; H_{xx}, H_{zz} and H_{zx} have two peaks, indicating the cavity has two pieces; particularly, the shape of H_{zx} is similar to half of the cross section of the cavity. The different sensitivity of components can be attributed to the stronger scattering from components normal to the cavity; y-polarized magnetic fields are tangential to both frustum and hemisphere and therefore cannot distinguish them; x-polarized magnetic fields are normal to the lower part of hemisphere and z-polarized magnetic fields are normal to the bottom of frustum, inducing larger responses in those areas.

To challenge the deep reading tools, we move the same cavity 11 m away from the borehole along the x axis; the cavity then fits into the fifth-level of the hierarchical sudoku mesh and only needs to be wrapped by one subdomain. A 18-m TR spacing is employed based on the borehole-object distance; the corresponding dominant components are plotted in Fig. 3.5. Let's first compare Fig. 3.5 with Fig. 3.4, both of which can distinguish the water saturated cavity from the oil saturated cavity well. Moreover, curves in Fig. 3.5 can be approximately extended from those

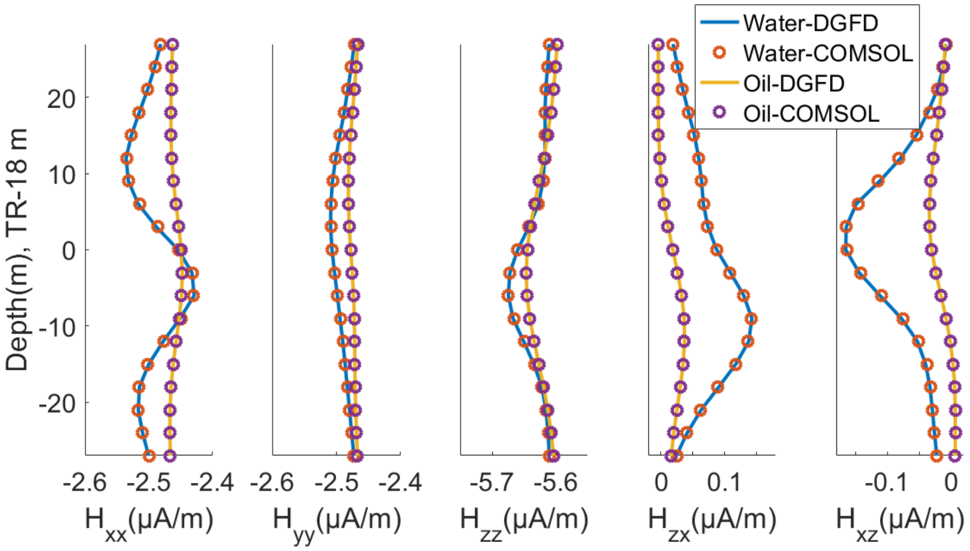


FIGURE 3.5: Dominant components of 18-m TR spacing are plotted to determine a cavity (17 m away from the borehole) saturated with water or oil.

in Fig. 3.4; the similarity indicates the target anomaly is the same, while the lower spatial resolution results from the tool's larger TR spacing to achieve a larger DOI. The normalized responses drop a lot, with H_{xx} around 3 % (0.5 %), H_{zx} around 6 % (1.5 %) and H_{xz} around 3 % (0.6 %) for the water (oil) saturated cavity.

Computational Analysis

At the linear equation solution phase, a two-step matrix solution procedure [Peng and Lee (2012)] is used to reduce a large linear system involving all the unknowns to an interface unknown system (3.1) involving only the interface unknowns. C_{ij} is the coupled interface unknown matrix between subdomains i and j; P_i can restrict the total DoFs to interface DoFs for subdomain i while P_i^T can revert it; L_i and U_i are the LU decomposed matrices of subdomain i; u_i and b_i are the interface unknowns and interface sources for subdomain i, respectively. To avoid the matrix-matrix division, we don't assemble the interface matrix explicitly, but use a Krylov subspace iterative method, the BiCGStab(L) [Sleijpen and Fokkema (1993)], to perform only matrix-vector multiplication (MVM) for coupled matrices and forward (backward)

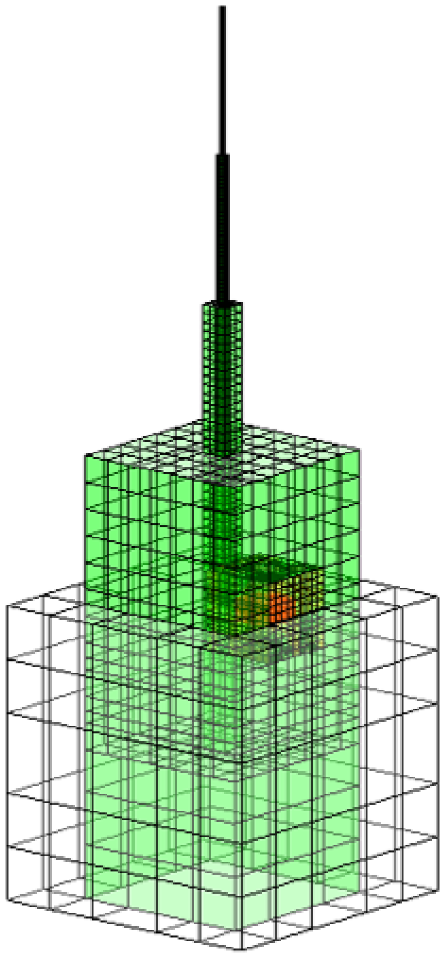
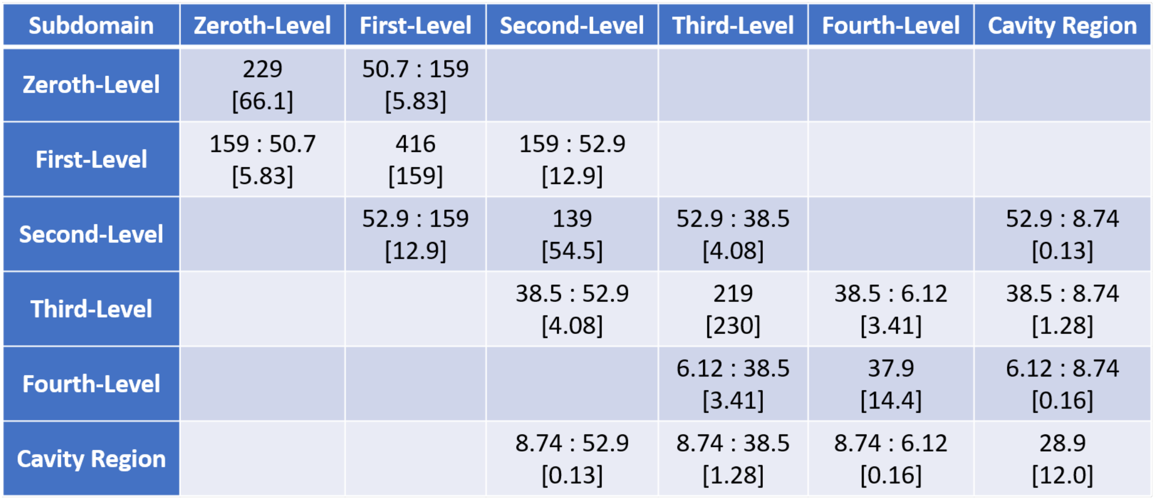


FIGURE 3.6: Hierarchical sudoku mesh to demonstrate the computational analysis

substitutions for L (U) matrices of all subdomains; the LU decompositions are only required to perform once per subdomain and computationally light since all the subdomain matrices not only have smaller DoFs due to the unconformal meshes, but also have block diagonality thanks to the finite basis functions; to overcome the convergence problem raised by the large complex eigenpairs of the linear system, L = 6 is employed for the BiCGStab(L), which requires 12 forward (backward)

Table 3.5: Matrix properties for the first deep reading case. Note that the data in the i^{th} diagonal entry represents: total DoFs ($\times 10^3$) [nnz of the L matrix ($\times 10^6$)] of the i^{th} subdomain; the data in the off-diagonal position ij represents: the interface DoFs of the subdomain $i : j$ ($\times 10^3$) [nnz of the coupled matrix C_ij ($\times 10^6$)]. Since the system is complex symmetric, only the diagonal and upper triangular part are shown.



Subdomain	Zeroth-Level	First-Level	Second-Level	Third-Level	Fourth-Level	Cavity Region
Zeroth-Level	229 [66.1]	50.7 : 159 [5.83]				
First-Level	159 : 50.7 [5.83]	416 [159]	159 : 52.9 [12.9]			
Second-Level		52.9 : 159 [12.9]	139 [54.5]	52.9 : 38.5 [4.08]		52.9 : 8.74 [0.13]
Third-Level			38.5 : 52.9 [4.08]	219 [230]	38.5 : 6.12 [3.41]	38.5 : 8.74 [1.28]
Fourth-Level				6.12 : 38.5 [3.41]	37.9 [14.4]	6.12 : 8.74 [0.16]
Cavity Region			8.74 : 52.9 [0.13]	8.74 : 38.5 [1.28]	8.74 : 6.12 [0.16]	28.9 [12.0]

substitutions per subdomain and 12 MVMs per coupled subdomain pair in each iteration.

$$\begin{pmatrix} I & P_1(L_1U_1)^{-1}P_1^TC_{12} & \cdots & P_1(L_1U_1)^{-1}P_1^TC_{1M} \\ P_2(L_2U_2)^{-1}P_2^TC_{21} & I & \cdots & P_2(L_2U_2)^{-1}P_2^TC_{2M} \\ \cdots & \cdots & \ddots & \vdots \\ P_M(L_MU_M)^{-1}P_M^TC_{M1} & P_M(L_MU_M)^{-1}P_M^TC_{M2} & \dots & I \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{pmatrix} \tag{3.1}$$

Since the multiplication number of both forward (backward) substitutions and the MVM are linear to the number of non-zero entries (nnz) of L (U) matrices and coupled matrices, respectively, the total number of multiplications can be obtained as

$$\mathrm{numMultiplication} = 12\times\mathrm{numLogging}\times\mathrm{numIter}\times[\sum_{\mathrm{i=1}}^{\mathrm{N}}(\gamma_{\mathrm{i}}\mathrm{nnzL_i} + \mathrm{nnzU_i}) + \sum_{\mathrm{i=1}}^{\mathrm{N}}\sum_{\mathrm{j=1}}^{\mathrm{N}}(\mathrm{nnzC_{i,j}})] \tag{3.2}$$

where numLogging is the number of the logging points, numIter is the iteration number of the BiCGStab(L), γ_i is positive correlated to the ratio between interface DoFs and total DoFs of subdomain i, $\mathrm{nnzL_i}$ ($\mathrm{nnzU_i}$) is the nnz of L_i (U_i), and $\mathrm{nnzC_{i,j}}$ is the nnz of C_{ij}; notice that since the total unknown system is complex symmetric, the $\mathrm{nnzC_{i,j}} = \mathrm{nnzC_{j,i}}$ and $\mathrm{nnzL_i} \approx \mathrm{nnzU_i}$. For coupled subdomains i, j, the right-hand-side (RHS) of the forward substitutions is padded with zero to its top by left multiplying P_i^T, increasing its length from the interface DoFs to the total DoFs, and therefore the intermediate solution will contain the same amount of zeros and the multiplication number is much less than the nnz of $\mathrm{L_i}$; the subsequent backward substitutions, however, do not benefit from the zeros of the RHS, because its substitutions start from the non-zero bottom part of RHS.

We take the example in Fig. 3.6 and analyze the computation complexity to solve it with the information listed in Table 3.5. By summing up the total DoFs of all subdomains, the total DoFs with hierarchical sudoku mesh are obtained as 1.07×10^6; to secure the same accuracy, COMSOL will need about 11.5×10^6 DoFs. The one order DoF reduction can be attributed to the spectral convergence of the high-order basis functions and the higher accuracy of structured or swept hexahedra over tetrahedra. Moreover, the actual unknown of the linear system to solve only involves the interface unknowns, which reduce to 0.315×10^6. The approximate number of multiplications for each iteration can also read from the Table 3.5 as $\sum_{\mathrm{i=1}}^{\mathrm{N}}(\gamma_{\mathrm{i}}\mathrm{nnzL_i} + \mathrm{nnzU_i}) + \sum_{\mathrm{i=1}}^{\mathrm{N}}\sum_{\mathrm{j=1}}^{\mathrm{N}}\mathrm{nnzC_{i,j}} \approx (562.8 + 27.8) \times 10^6$, where we suppose $\gamma_{i,j} = 0.05$ for all coupled subdomains; the large difference between these two nnz indicates the forward (backward) substitution dominates the computational burden of the iterative solver. In use of (3.2), the number of the multiplications will be around 1.8×10^{12}, given numLogging = 24×3 for 24 logging points with 3 polarizations and average numIter = 3.5 for a relative residue around 3×10^{-6}. The computational time shows the DGFD code beats COMSOL for the Deep reading I with 100 min to 600 min.

3.1.4 Summary

A hierarchical sudoku mesh is introduced to model the induction logging problem efficiently based on the DGFD. It inherits the gradient property from the progressive tetrahedron mesh and the high-order structured mesh, and then combines them by the DGFD-based DDM. Because the hierarchical sudoku mesh's specific spatial discretization conforms to the diffusion pattern and the geometrical characteristics of the logging problem with as few unknowns as possible, it provides both fast convergence and short running time in each iteration step. A formation evaluation problem is used to verify the accuracy of the proposed hierarchical sudoku mesh, and a deep reading induction tool is employed to distinguish a cavity saturated with water from oil 6 m (17 m) away from the borehole in carbonate environment successfully. Both cases demonstrate that with the help of hierarchical sudoku mesh, the DGFD-based DDM, and theoretically all DDM schemes, can have a better performance when modeling borehole logging problem over the traditional FEM.

3.2 Flipped TF/SF Mixed Solver for LWD Problems

This section presents a fast and memory-efficient 3D electromagnetic solver for LWD tools based on the flipped TF/SF DGFD method. The new method inherits the RTC with surface current sources from the mixed TF/SF DGFD method to couple non-conformal meshes as well as TF/SF solvers; it then extends to the moving tool scenarios for the LWD application with a solver flipping technology and delivers a single global matrix of dramatically reduced dimension for all source positions, which can be solved directly on a normal-memory computer. By incorporating the tetrahedra/hexhedra mixed mesh and the curved domain decomposition method, the new solver is applied to two LWD cases and about 70% unknowns can be saved for the flipped TF/SF DGFD remeshed regions.

3.2.1 Background

Electromagnetic LWD tools [Pardo et al. (2006); Wang et al. (2006); Hong et al. (2014); Pardo and Torres-Verdín (2015); Chen (2017)] have been widely used in the oil and gas industry to improve the efficiency of the well placement; through measuring the formation resistivity, both porosity and hydrocarbon saturation can be estimated. The new generation of LWD tools, such as the deep directional resistivity (DDR) tool, can provide an improved depth of investigation DOI of over 30 m, and then can better map the near-wellbore reservoir structure [Hu et al. (2017)], near-wellbore hydraulic fractures [Fang et al. (2017); Dai et al. (2018)] and navigate the wellbore trajectories [Chaumont-Frelet et al. (2018)]. To interpret the DDR measurements in high-angle and horizontal (HA/HZ) wells, however, is very different from wireline logging modeling technologies for vertical wells; a full 3D inversion method [Yu et al. (2017)] is required to process the DDR measurements and reconstruct the 3D distribution of electrical conductivity in real time [Puzyrev et al. (2018)]. Nevertheless, one of the major difficulty of the 3D inversion is the high computational burden, which can be attributed to the involved large number of forward modelling for the moving tool (different transmitter-receiver positions in a long HA/HZ well). Therefore, a fast 3D forward solver [Ma et al. (2014b,a); Ren et al. (2017); Zeng et al. (2018a); Zhang et al. (2019a)] is demanded to model this problem.

To efficiently model the LWD tool, the most intuitive way is to include all transmitter-receiver positions by a single mesh [Wang et al. (2019)]; then the discretization of the governing equation will generate one sparse linear system with multiple RHSs, which can be solved by a parallel direct solver [da Silva et al. (2012); Puzyrev et al. (2016)] at a comparable cost to one matrix factorization. This elegant approach, however, can easily generate a large amount of unknowns, because: 1. the long wellbore trajectory requires to be meshed densely to guarantee the computational accuracy, because the directly imposed dipole sources lead to the field singularity and requires a large amount of elements near the sources to alleviate; 2. a large area of formation background needs to be included because the large DOI of DDR tools. As the direct solver is much more memory demanding than the iterative solver [Puzyrev et al. (2013); Grayver and Bürg (2014)], several million unknowns (the specific number depends on both the model and the solver) can be hard to solve on a single computer with 128 GB memory (one of the most common configurations for a high-performance computing node). To fit the sparse linear system of the 3D LWD model into the direct solver on a single computer, or more generally, to increase the computing capability under the same computing resource, a flipped TF/SF DGFD method is employed to reduce the dimension of the linear system. The TF/SF DGFD method is first proposed in [Sun et al. (2018)] for fixed-source subsurface problems based on hexahedra and an iterative solver; the SF DGFD framework is employed for subdomains with sources and the TF DGFD framework is employed for the remaining subdomains. The advantage of this TF/SF decomposition strategy is: 1) the SF solver can model the source region more accurately than the TF solver given the same mesh by removing the field singularity; 2) the source is always wrapped by a homogeneous subdomain with the SF solver and all other regions employ the TF solver; consequently, the equivalent source can be added at the interface between the TF-based subdomain and the SF-based subdomain rather than added over the scattering objects as the pure SF solver does, and correspondingly the time-consuming elementwise volume integral is transformed into the cheap surface integral. This decomposition strategy, however, is not suitable and not comprehensive to model the moving tool in LWD applications. First, the anomaly region is usually employed with dense meshes to capture its geometry feature; therefore, when the source locates inside this region, the TF solver should be sufficient to guarantee a good accuracy and the SF solver is then redundant. Second, the workflow of the domain decomposition and the solver distribution is not provided for the moving tool, and most importantly, whether the TF/SF solver can keep the invariance of the system matrix for the moving tool is not studied yet. Third, if the system matrix is invariant for different source positions, then a direct solver should be used instead of the iterative solver to handle multiple RHSs all at once. Fourth, the subsurface can be very distorted and the layered formation can be curved, which requires to incorporate other techniques into the TF/SF DGFD framework to model complicated scenarios.

To extend the TF/SF DGFD to the LWD application and answer above four questions, we propose a flipped TF/SF DGFD method in Section 3.2.2, which not only keeps the advantage of avoiding elementwise integration of the scattering objects, but also handles the moving source scenario efficiently and elegantly; the modified domain decomposition strategy is then demonstrated to share the same system matrix for different source positions in Section 3.2.2; tetrahedra/bricks mixed meshes are introduced to efficiently model a cylindrical lithologic reservoir w/ or w/o a transition zone in Section 3.2.3; the tetrahedra/hexahedra mixed meshes and curved domain decompositions are then introduced to model a large-scale oil trap case including a salt dome, a fold and a fault in Section 3.2.3. For all cases, the parallel sparse direct solver intel MKL PARDISO (called as PARDISO in the following) is employed to solve multiple RHSs together efficiently [Schenk and Gärtner (2004)].

3.2.2 Flipped TF/SF DGFD Method

New Domain Decomposition Workflow

Based on the time harmonic assumption with the $e^{j\omega t}$ time dependence, both the TF-based and SF-based wave equations can be written as the unified formula

$$\begin{aligned} &\frac{1}{jw\mu_0}\nabla \times (\mu_r^{-1}\nabla \times \mathbf{E_{t/s}}) + jw\epsilon_0\epsilon_r\mathbf{E_{t/s}} \\ &\quad = -\mathbf{J_{i/eq}} - \frac{1}{jw\mu_0}\nabla \times (\mu_r^{-1}\mathbf{M_{i/eq}}) \end{aligned} \tag{3.3}$$

where $\mathbf{E_t}$ is the total electric field radiated from imposed electric dipoles $\mathbf{J_i}$ and imposed magnetic dipoles $\mathbf{M_i}$, and $\mathbf{E_s}$ is the scattered electric field due to equivalent electric dipoles $\mathbf{J_{eq}} = jw\epsilon_0(\epsilon_r - \epsilon_{rb})\mathbf{E_b}$ and equivalent magnetic dipoles $\mathbf{M_{eq}} = jw\mu_0(\mu_r - \mu_{rb})\mathbf{H_b}$. The background fields $\mathbf{E_b}$ and $\mathbf{H_b}$ interact with the object to serve as the equivalent sources. $\epsilon_0(\mu_0)$, $\epsilon_r(\mu_r)$, $\epsilon_b(\mu_b)$ are the vacuum permittivity (permeability), material and background relative permittivity (permeability), respectively.

With the Galerkin's method, the curl-conforming basis functions $\mathbf{\Phi}$ are employed for testing. Assuming that the computational domain is divided into N nonoverlapping subdomains, the weak form of (3.3) is given for the i th subdomain as:

$$\begin{aligned} &\frac{1}{jw\mu_0}\int_{\Omega^{(i)}} \nabla \times \mathbf{\Phi}^{(i)} \cdot {\mu_r^{(i)}}^{-1}\nabla \times {\mathbf{E_{t/s}}}^{(i)} dV \\ &+ jw\epsilon_0\int_{\Omega^{(i)}} \mathbf{\Phi}^{(i)} \cdot \epsilon_r^{(i)}{\mathbf{E_{t/s}}}^{(i)} dV - \oint_{\partial\Omega^{(i)}} \mathbf{\Phi}^{(i)} \cdot \hat{n}^{(i)} \times \mathbf{H}^*_{\mathbf{t/s}} dS \\ &= -\int_{\Omega^{(i)}} \mathbf{\Phi}^{(i)} \cdot {\mathbf{J_{i/eq}}}^{(i)} dV - \frac{1}{jw\mu_0}\int_{\Omega^{(i)}} \nabla \\ &\quad \times \mathbf{\Phi}^{(i)} \cdot {\mu_r^{(i)}}^{-1}{\mathbf{M_{i/eq}}}^{(i)} dV \end{aligned} \tag{3.4}$$

For LWD applications, the TF solver will require dense meshes around both sources and anomalies; while the anomalies indeed demand dense meshes to capture complex geometries, the excessive meshing near the sources can be alleviated by employing the SF solver [Puzyrev et al. (2018); da Silva et al. (2012); Grayver and Bürg (2014); Sun and Nie (2008)]. To further save unknowns, coarse meshes with high-order basis functions [Sun and Nie (2008)] can be employed to capture the smooth scattered fields more efficiently.

However, (3.5) shows that the equivalent sources of the SF solver add nontrivial computations as integrating over scattering objects as many times as the number of source positions. Even worse, the layered formation can make the volume of scattering objects comparable to the whole computational domain. To tackle the undesired integrals, a new domain decomposition strategy is proposed here for LWD applications:

1. A single mesh grid and a single domain decomposition are employed for different source positions to reuse the system matrix;

2. Anomalies require dense tetrahedra to capture complicated or distorted structures; therefore, the TF solver can be employed for these subdomains naturally without introducing the field singularity.

3. Planar or curved layered formations can be discretized effectively by hexahedra from sweeping quadrilaterals;

    (a) For each single source, the subdomain that contains it should use the SF solver to avoid field singularity, and also be homogeneous to avoid the undesired volume integrals; other subdomains can contain different media but require the TF solver to avoid volume integrals.

(b) To model all the sources with a single mesh, any subdomain that ever uses the SF solver should be homogeneous; in other words, each layer through which the tool penetrates should be distributed into independent subdomains.

(c) Since subdomains are either using the SF solver or away from the source, the computed fields are supposed to be smooth, where high-order basis functions fit well with fewer unknowns.

The solver distribution can be summarized as follows: the subdomains of anomalies and the subdomains of formations that the tool does not penetrate through always employ the TF solver; the subdomains of formations that the tool penetrates through will flip between the TF and SF solver for different source positions. Since subdomains are not assigned with a single solver but may flip between two solvers, we call this new strategy the flipped TF/SF DGFD method.

Invariance of the System Matrix

To use a single matrix for all sources in the LWD problem, we need to make sure the flipping of the TF/SF solver honors the invariance of the system matrix. Let's consider a unified weak form of wave equation for the TF/SF solver in nonoverlapping subdomain i as,

$$\begin{aligned}
&\frac{1}{jw\mu_0}\int_{\Omega^{(i)}} \nabla\times\mathbf{\Phi}^{(i)}\cdot{\mu_r^{(i)}}^{-1}\nabla\times{\mathbf{E_{t/s}}}^{(i)}dV \\
&+\ jw\epsilon_0\int_{\Omega^{(i)}}\mathbf{\Phi}^{(i)}\cdot\epsilon_r^{(i)}{\mathbf{E_{t/s}}}^{(i)}dV - \oint\limits_{\partial\Omega^{(i)}}\mathbf{\Phi}^{(i)}\cdot\hat{n}^{(i)}\times\mathbf{H}^{*}_{\mathbf{t/s}}dS \\
&= -\int_{\Omega^{(i)}}\mathbf{\Phi}^{(i)}\cdot{\mathbf{J_{i/eq}}}^{(i)}dV - \frac{1}{jw\mu_0}\int_{\Omega^{(i)}}\nabla \\
&\quad\times\ \mathbf{\Phi}^{(i)}\cdot{\mu_r^{(i)}}^{-1}{\mathbf{M_{i/eq}}}^{(i)}dV
\end{aligned}\tag{3.5}$$

where $\mathbf{\Phi}$ is the curl-conforming testing functions [Nédélec (1980)], $\mathbf{E_t}$ is the total electric field radiated from imposed electric (magnetic) sources $\mathbf{J_i}$ ($\mathbf{M_i}$), and $\mathbf{E_s}$ is the scattered electric field from equivalent electric (magnetic) sources $\mathbf{J_{eq}} = jw\epsilon_0(\epsilon_r - \epsilon_{rb})\mathbf{E_b}$ ($\mathbf{M_{eq}} = jw\mu_0(\mu_r - \mu_{rb})\mathbf{H_b}$). The background fields $\mathbf{E_b} = \mathbf{E_t} - \mathbf{E_s}$ and $\mathbf{H_b} = \mathbf{H_t} - \mathbf{H_s}$ interact with the object to serve as the equivalent sources. $\epsilon_0(\mu_0)$, $\epsilon_r(\mu_r)$, $\epsilon_b(\mu_b)$ are the vacuum permittivity (permeability), material and background relative permittivity (permeability).

If the computational domain has not been divided into subdomains, the surface integral term in (3.5) will be replaced by appropriate outer boundary conditions like the ABC or the perfectly matched layer (PML); then it is clear that a single bilinear term and system matrix will be shared for different sources and different solvers. However, the TF/SF DGFD method introduces a modified RTC between the subdomains of different solvers, differing from the normal one by a current source term as detailedly shown in [Sun et al. (2018)]. Whether this will alter the invariance of the system matrix under solver flipping has not been studied and will be tackled here. By supposing i as the local subdomain and j as the adjacent subdomain, we unify the RTC of the numerical magnetic flux for two different TF/SF solver combinations (i: TF solver & j: SF solver, and i: SF solver & j: TF solver)

$$\begin{aligned}\hat{n}^{(i)}\times\mathbf{H}^{*}_{\mathbf{t/s}} = {} & \left(Z^{(i)} + Z^{(j)}\right)^{-1} \times [Z^{(i)}\hat{n}^{(i)} \times \mathbf{H_{t/s}}^{(i)} \\ & + Z^{(j)}\hat{n}^{(i)} \times (\mathbf{H_{s/t}}^{(j)} \pm \mathbf{H_b}) \\ & + \hat{n}^{(i)} \times \hat{n}^{(i)} \times (\mathbf{E_{t/s}}^{(i)} - (\mathbf{E_{s/t}}^{(j)} \pm \mathbf{E_b}))]\end{aligned} \tag{3.6}$$

where $\hat{n}^{(i)}$ denotes the outward normal unit vector from subdomain i at boundaries, $(\cdot)^*$ denotes the theoretical field at the boundaries. $Z^{(i)}$ and $Y^{(i)}$ are the complex impedance and admittance in subdomain i. From (3.6) we read, when the local subdomain i uses the TF solver and the adjacent subdomain j uses the SF solver, the numerical flux generated by the scattered fields from subdomain j need to add the background fields to ensure the energy conservation; conversely, when the local subdomain i uses the SF solver and the adjacent subdomain j uses the TF solver, the numerical flux generated by the total fields from subdomain j need to subtract the background fields to ensure the energy conservation.

To demonstrate that the additional current source term will only appear as a RHS term in the linear system and will not alter the system matrix, we plug (3.6) into (3.5) and only keep the surface integral term, the equation will then become

$$\begin{aligned}&\oint\limits_{\partial\Omega^{(i)}} dS\mathbf{\Phi}^{(i)} \cdot \left(Z^{(i)} + Z^{(j)}\right)^{-1} \\ &\times \left[Z^{(i)}\hat{n}^{(i)} \times \mathbf{H_{t/s}}^{(i)} + Z^{(j)}\hat{n}^{(i)} \times \mathbf{H_{s/t}}^{(j)}\right. \\ &\left.+ \hat{n}^{(i)} \times \hat{n}^{(i)} \times (\mathbf{E_{t/s}}^{(i)} - \mathbf{E_{s/t}}^{(j)})\right] \\ = &\oint\limits_{\partial\Omega^{(i)}} dS\mathbf{\Phi}^{(i)} \cdot (\mp Z^{(j)}\hat{n}^{(i)} \times \mathbf{H_b} \pm \hat{n}^{(i)} \times \hat{n}^{(i)} \times \mathbf{E_b})\end{aligned} \tag{3.7}$$

The left-hand-sides (LHSs) will contribute to a system matrix the same as a normal RTC, since $\mathbf{E_{s/t}}, \mathbf{H_{s/t}}$ will be merged into the field unknowns and not affect the system matrix; the RHS is the additional term enforced by the energy conservation, which can also be viewed as a surface current source because no unknowns are involved. Recall that the flipped TF/SF DGFD method ensures that only the formation subdomain with sources employ the SF solver: from the adjacent TF-based subdomains' view, the fields from imposed source are "delayed" until the surface current source, which effectively remove the field singularity from the imposed dipole sources; from the SF-based subdomain's view, the background field information from the TF-based domains is filtered out by the surface current source and only the scattered fields information is fed back. The RTC of the numerical electric flux can get the same conclusion and will not be repeated here.

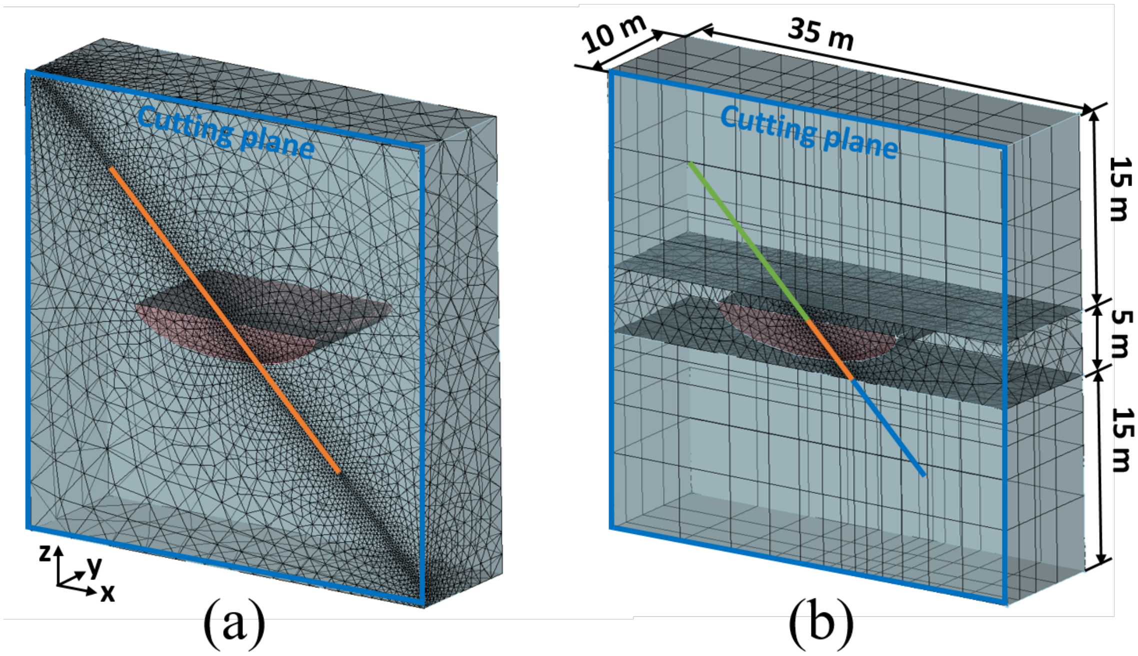


FIGURE 3.7: Comparison between the conformal tetrahedra and non-conformal tetrahedra/bricks mixed meshes for the infinitely long lithologic reservoir model

3.2.3 Numerical Results and Discussions

Cylindrical Lithologic Reservoir Model w/ or w/o a Transition Zone

We start from an infinitely long lithologic reservoir model similar to that in Section IV-A of [Wu et al. (2019)]. For one thing, this is a 2D model, for which the 2.5D fast algorithm proposed in [Wu et al. (2019)] can validate the flipped TF/SF DGFD method; for another, we can then taper the reservoir along the invariant direction and demonstrate the full 3D model's influence on both the conventional LWD tool and the DDR tool.

As shown by Fig. 3.7, an infinitely long cylindrical reservoir (pink region, 1 S/m conductivity) with a semi-elliptical cross section of 8-m major axis and 3-m minor axis lies along the y direction in a homogeneous formation (light blue region, 0.01 S/m conductivity). The computational domain is a 35 m × 20 m × 35 m box, half of which is depicted here to expose the 20-m long logging path with a 45° deviation angle (orange solid line) on a center section vertical to the y direction. The TF solver (Fig. 3.7a) requires dense tetrahedra (0.2-m mesh size) along the logging path to accurately model the fields near sources. In contrast, the flipped TF/SF DGFD method divides the whole computational domain into three subdomains (Fig. 3.7b); the middle anomaly subdomain keeps the same mesh, while the top and bottom formation subdomains employ structured meshes with high-order basis functions to save the total unknowns. Notice that if we notate the solvers for different subdomains in the order of top/middle/bottom, we will have SF/TF/TF for the source located at the top subdomain (green line), TF/TF/TF for the source located at the middle subdomain (orange line) and TF/TF/SF for the source located at the bottom subdomain (blue line). The meshes of the top and bottom subdomains become coarser away from the middle subdomain, because all the sources are either directly imposed inside the middle subdomain, or added as the surface current sources at the middle subdomain's top/bottom boundary.

The logging curves of a conventional LWD tool working at 100 kHz with a 8-feet TR spacing are computed by the 2.5D FDM, the 3D FEM with conformal tetrahedra and 3D TF/SF DGFD as shown in Fig. 3.8. The label of y-axis represents the true vertical depth (TVD) of the midpoint between the transmitter and the receiver, with the TVD of the reservoir's horizontal plane as zero; besides, the component direction is denoted by tool coordinates. Notice that for the DGFD results, all the receivers in the SF-based subdomain have already added a background field; the background field is the same for different receivers because the formation background is homogeneous and the TR spacing is fixed. Fig. 3.8 shows good agreement for the complex coplanar components among three methods. The 2.5D FDM computes a sliding 12 m × 12 m square domain for every logging point and employs excessive meshes (16,926 DoFs) to correctly capture the curved reservoir to serve as the reference. The 3D FEM

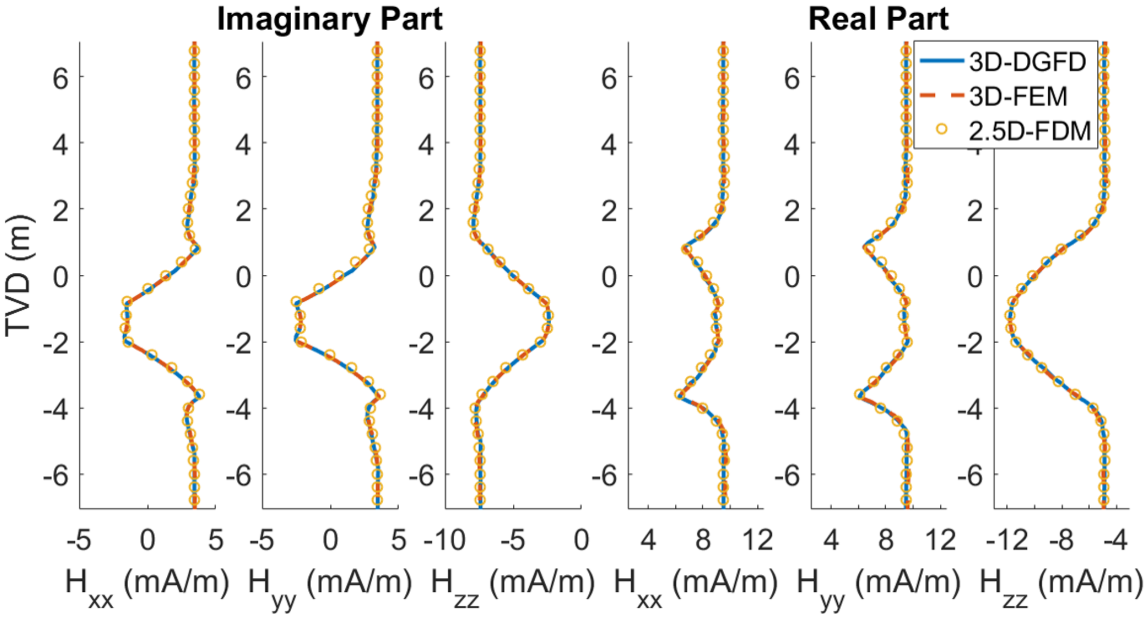


FIGURE 3.8: Comparison of all three complex coplanar components among the 3D-DGFD, 3D-FEM and 2.5D-FDM

discretizes the whole computational domain as shown in Fig. 3.7a with tetrahedra of the second-order basis functions; the PARDISO complex and symmetric matrix solver requires 25 GB and 3 minutes on a single Intel Xeon E5-2680v3 processor of 12 cores to numerically factorize its linear matrix of 1,306,972 DoFs. The 3D DGFD divides the computational domain into three subdomains as shown in Fig. 3.7b, and models the middle subdomain with the same tetrahedra as its counterpart in the 3D FEM modeling and the top/bottom subdomains with bricks of the fifth-order basis functions; the dimension of the resultant system decreases to 653,420 (295,890 for the middle subdomain and 178,765 for the top(bottom) subdomain), and can be factorized by the PARDISO solver with 17 GB and 2 minutes on the same processor. Note that if we only compare the TF/SF DGFD remeshed region, the DoFs reduces from about 1,000,000 to 357,530, which saves up to 65% unknowns. Additionally, one may notice that the decrease of the running time and memory cost is not as large

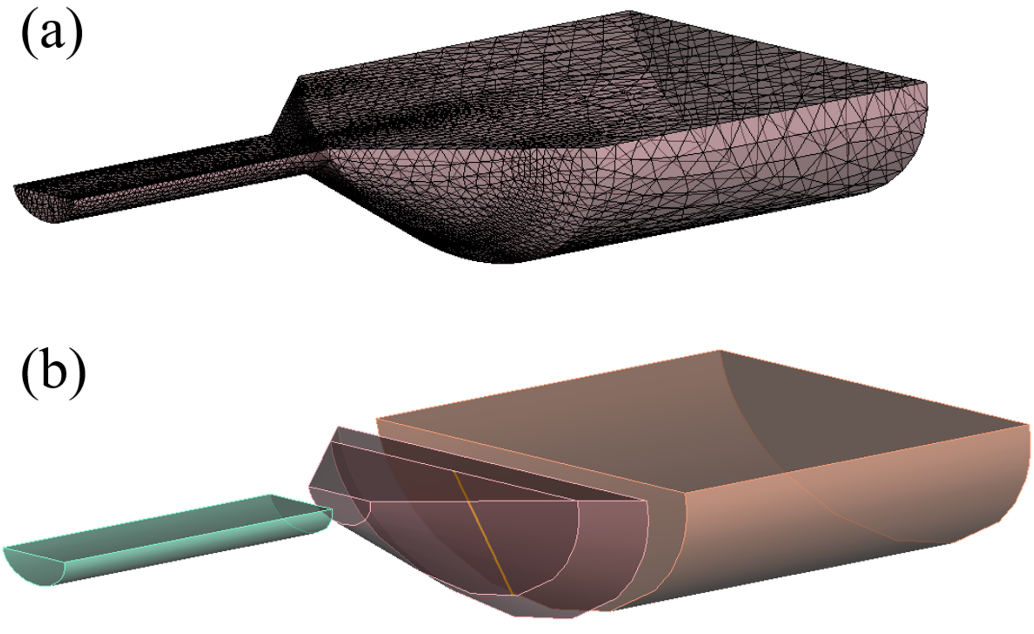


FIGURE 3.9: (a) The tetrahedral mesh for the lithologic reservoir with a transition zone; (b) The lithologic reservoir is split into three parts: a small-scale cylindrical reservoir (aquamarine), a conical transition zone (light pink), and a large-scale cylindrical reservoir (light salmon).

as the reduction of the unknowns, because the DGFD generated matrix is denser due to the surface integral from the RTC and the high-order basis functions.

However, if the reservoir is not invariant along the y direction, but tapers, the model could be over simplified by using the 2.5D algorithm. To investigate this problem, we move the working plane of the LWD tool to a transition zone, which connects two cylindrical lithologic reservoirs with different dimensions as shown in Fig. 3.9. The length of major axis and minor axis of the small-scale (large-scale) reservoir is 2 m (10 m) and 0.75 m (3.75 m), respectively; the length of the transition zone is 4 m. To make a fair comparison against the previous case, we specifically choose an elliptical cross section of 8-m major axis and 3-m minor axis as the working plane; the logging path penetrating the reservoir is highlighted by an orange line in

Fig. 3.9(b). The tilted bottom face of the transition zone introduces some disturbance to the logging curves as shown in Fig. 3.10 around -2.8 m (the TVD of the intersection between the logging path and the tilted bottom face of the transition zone), where the relative errors due to the 2D simplification are above 10%, 12% and 10% for the H_{xx}, H_{yy} and H_{zz} components, respectively. Except for the large error around the tilted face, the 2D model successfully captures the major characteristic of the logging curve, attributed to the shallow detection depth of the conventional LWD tool.

However, if we employ a DDR tool of 50-feet TR spacing and 12 kHz working frequency, the 3D nature of the reservoir transition zone will be more evident, which is verified in Fig. 3.11. The H_{yy} component has the largest difference (25%) from the 2D simplification, because it lies along the transition zone's extending direction.

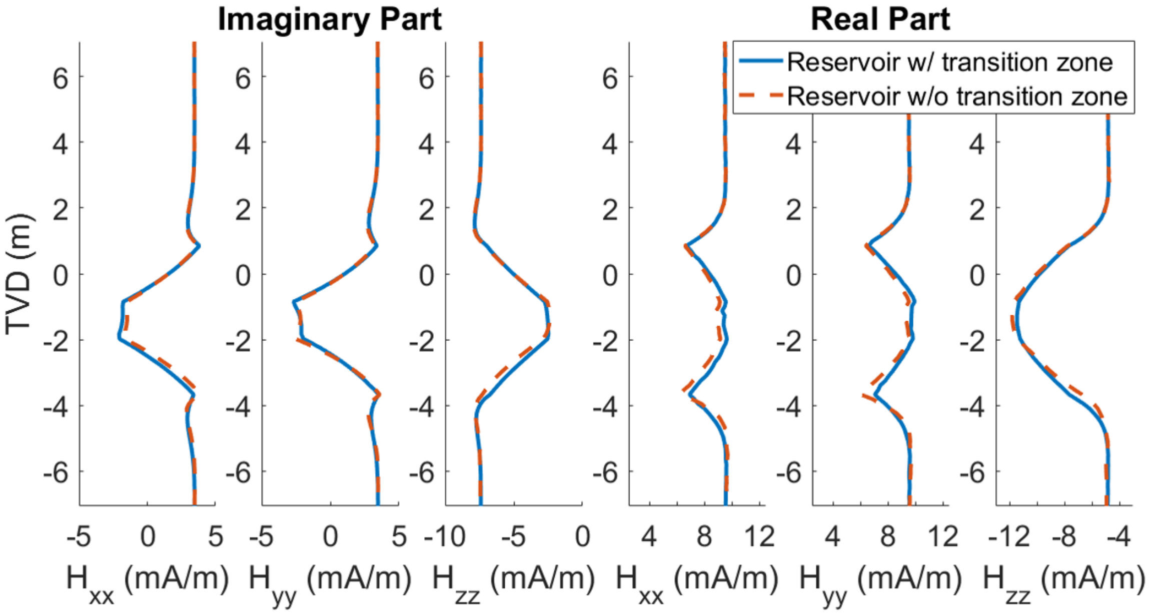


FIGURE 3.10: Comparison of all three complex coplanar components between the reservoir w/ and w/o the transition zone from the LWD tool of a 8-feet TR spacing

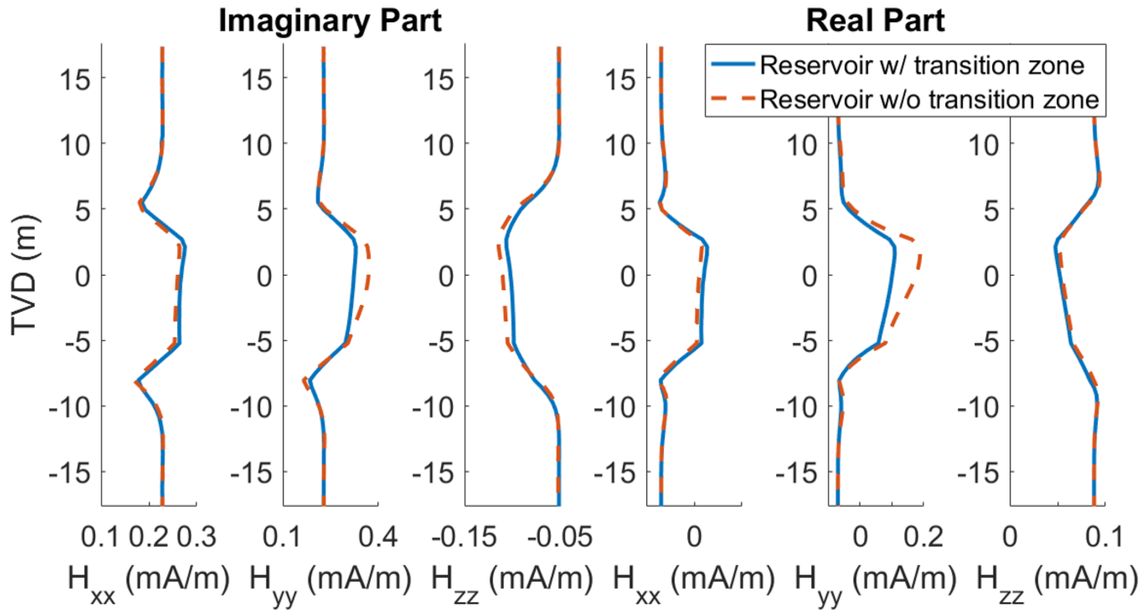


FIGURE 3.11: Comparison of all three complex coplanar components between the reservoir w/ and w/o the transition zone from the DDR tool of a 50-feet TR spacing

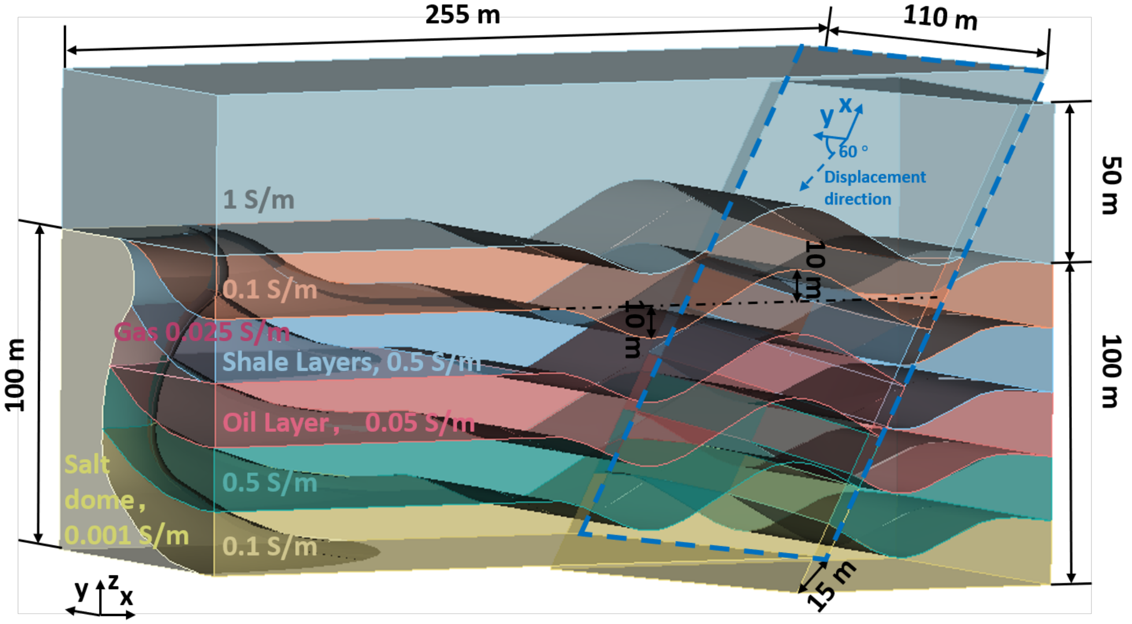


FIGURE 3.12: A large-scale oil trap model including a salt dome, a fold and a fault

3D Oil Trap on Salt Dome Flanks Model

To show the computational advantage of the proposed memory-efficient LWD solver, we consider a large-scale complicated oil trap model, which starts from the flank of a salt dome and extends to a fold/fault mixed region as shown in Fig. 3.12. The thickness of each layer is 20 meters, except that of the top layer is 50 meters. The salt dome modeled here is a quadrant of a 100 m high roof of a large salt dome; it raises the adjacent layers and therefore creates a natural gas trap (pink) at its flank. The natural gas stands on an oil layer (coral), and the oil layer extends to a fold/fault mixed region. The fold includes two synclines and one anticline, which share the same 350-m curvature radius and 10-m maximum deviation from the original formation plane. The ulterior syncline suffers from a high-angle reverse dip-slip fault, whose fault plane (blue dashed parallelogram) can be obtained from a xy plane, by first rotating -60° about the y axis and then 30° about the x axis; the direction of the 15-m displacement (blue dashed arrow) is 60° from the rotated y direction (blue coordinates) of the fault plane. The specific electric conductivity of each layer (numbered from top to bottom) is listed in Table 3.6.

Table 3.6: Parameters for the oil trap case

Material	**Conductivity (S/m)**
Layer I	1
Layer II & VI	0.1
Layer III (Shale) & V	0.5
Layer IV (Oil)	0.05
Gas trap	0.025
Salt dome	0.001

The tetrahedra meshed 110 m × 300 m × 150 m computational domain is shown in Fig. 3.13 by exposing the tool's compound logging path (orange line), including a 45° tilted section adjacent to the salt dome in working plane I (45° rotated from the yz plane about the salt dome's axis, white solid parallelogram), a horizontal section and another 45° tilted section, both penetrating the oil and shale layers in working plane II (the center section vertical to the y direction, black dashed parallelogram). From Fig. 3.13 we can observe that: 1) tetrahedra concentrate around the logging path to attain good accuracy; 2) structures in the vicinity of the salt dome (called as the salt dome region in the following) are sharper and more distorted, where the tetrahedra can capture the geometry much easier than the hexahedra; 3) shapes are less distorted and sweepable in the fold/fault mixed region, where hexahedra are more suitable if mesh refinement around the logging path is not required. By following the same spirit as above, we avoid the excessive meshes around the logging path in the less distorted region (the oil layer and shale layer in the fold/fault region) by employing the SF solver locally; then the hexahedra can be used in the fold/fault

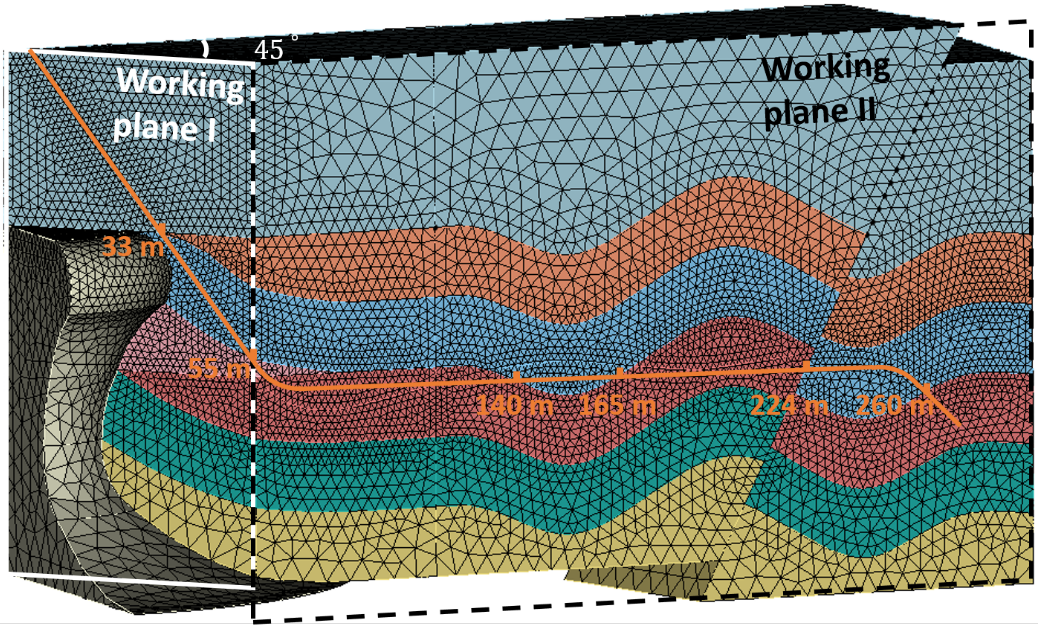


FIGURE 3.13: Large-scale oil trap model is meshed by conformal tetrahedra; one logging path (orange solid line) and two working planes (white solid line and black dashed line) are exposed by hiding half of the model.

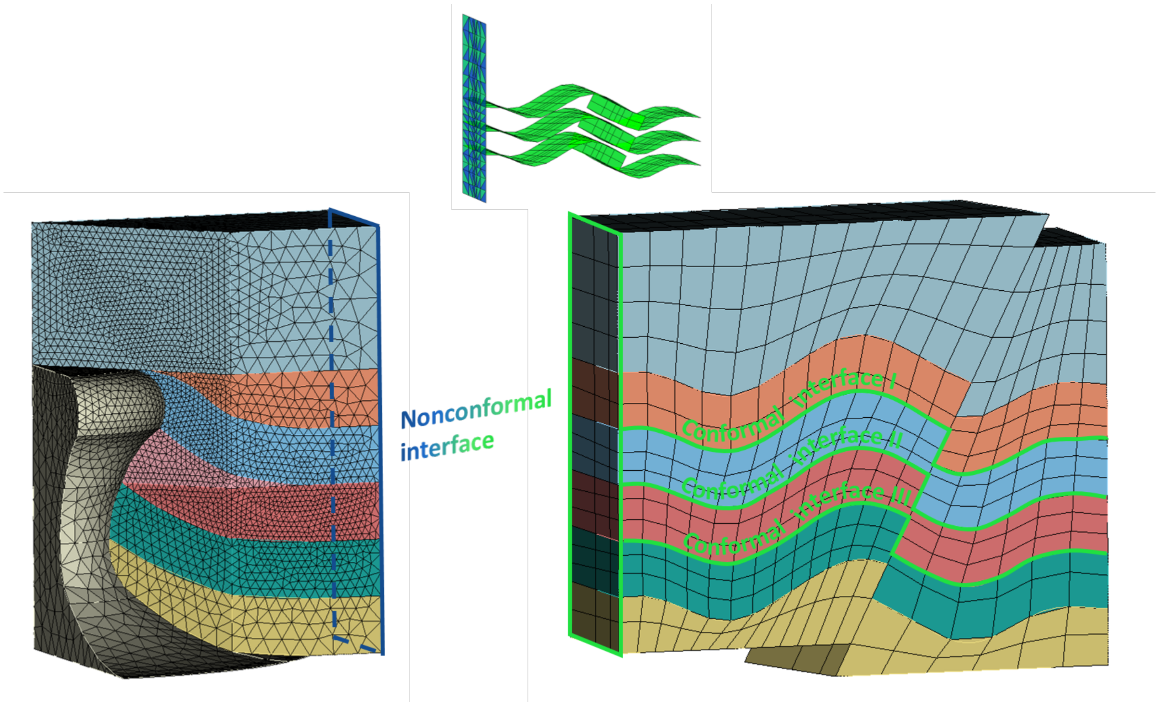


FIGURE 3.14: Nonconformal tetrahedra/hexahedra mixed meshes for the large-scale oil trap model. The inset exposes all the domain decomposition interfaces for this case, including one nonconformal interface (blue/green surface) between the tetrahedra meshed salt domain region and the hexahedra meshed fold/fault region, and three conformal interfaces (green surfaces) inside the hexahedra meshed fold/fault region.

region, with a nonconformal domain decomposition from the salt dome region. Further, to avoid the repetitive volume integral over the equivalent source (curved layers with conductivity different from the source embedded layer), the fold/fault region needs to be further divided into four subdomains: the oil layer and the shale layer are separated into independent subdomains to guarantee the subdomain homogeneity for the SF solver when the tool is penetrating through; the two top (bottom) layers can be combined to one subdomain since the TF solver will always be used for them. The tetrahedra/hexahedra mixed meshes and corresponding domain decomposition interfaces are plotted in Fig. 3.14.

Both the real and imaginary part of the coplanar components of a DDR tool

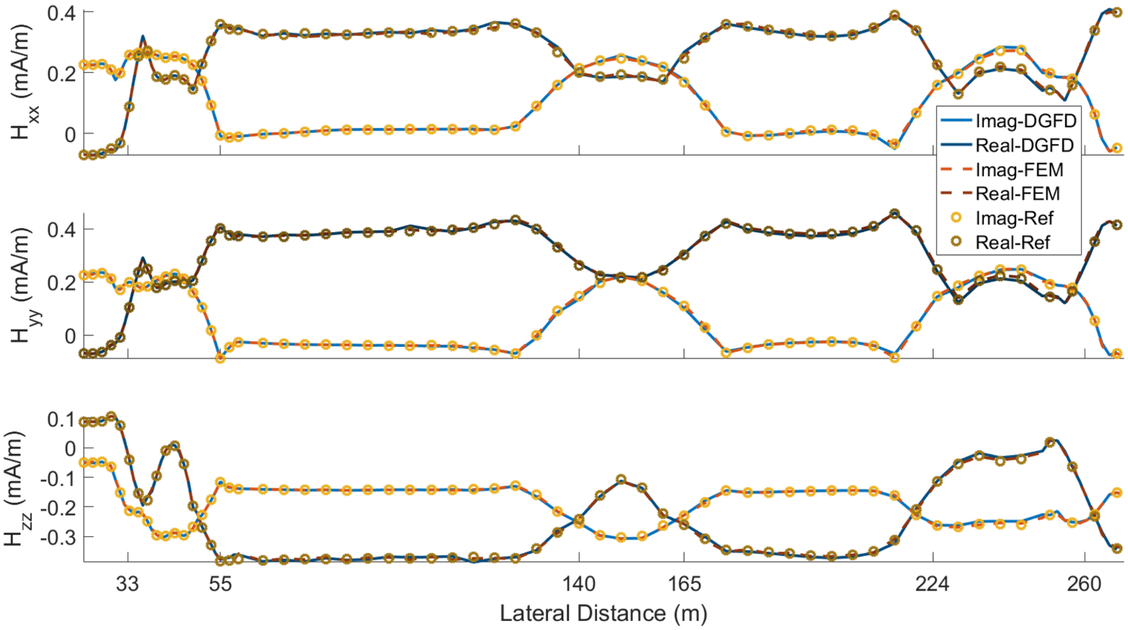


FIGURE 3.15: Complex coplanar component comparison for the large-scale oil trap model among the TF/SF DGFD, the FEM with 2-m smallest meshe size, and the FEM with 1-m smallest mesh size as the reference

(50-feet TR spacing, 12 kHz) are plotted in Fig. 3.15, with the true horizontal length as the x axis. Note that for the DGFD results, all receivers at SF-based subdomains have already added background fields; although the logging direction is variant, this addition is straightforward, since all components of the computed scattered fields and the analytical background fields are in tool coordinates. The conventional tetrahedra-based FEM with extremely dense meshes (1,767,624 elements with 1-m smallest mesh and 11,235,456 DoFs) serves as the reference and the solution with the PARDISO occupies more than 400 GB on a large-memory Haswell node [Towns et al. (2014)]. After refining mesh for both the FEM and the flipped TF/SF DGFD method from an initial coarse mesh, we can achieve around 2% relative errors as shown in Fig. 3.15, where the FEM uses about 438,243 tetrahedra (2-m smallest mesh) and the DGFD method uses 185,783 tetrahedra and 2,040 hexahedra (from top to bottom 768,360,360,552) of the fourth-order basis function. Consequently, the

DGFD method reduces the total DoFs (the numerical factorization memory cost) from 2,800,330 (67 GB) to 1,627,140 (35 GB), with a speedup from 12 minutes to 4 minutes. For the remeshed fold/fault region, the DoF decreases from about 1,600,000 to only 452,340, saving up to 70% unknowns. Since the direct solver is very memory demanding and the required memory is superlinear to the DoFs, the new method can increase the computing capability for the LWD application greatly. For anisotropic formations, the background fields will be numerically integrated instead of closed-form, which requires to solve transverse electric and magnetic fields in the spectral domain through numerical integral and then transform fields into the spatial domain as detailed in Section II-A of [Hu et al. (2018)].

Let's interpret the logging curves in Fig. 3.15 a little more: when the tool penetrates through the uplifted layers near the salt dome at a 45° deviation angle, large responses are captured at around 33 m; then, the tool passes through the gas trap (55 m), enters the oil layer and turns to the horizontal direction, along with which the magnetic fields vary sharply and then becomes smooth; after the first horizontal navigation stage inside the oil layer, the tool enters the shale layer from 140 m to 165 m due to the fold, which induces an obvious variation of the logging curves; after the tool comes back to the oil layer for the second horizontal stage, it then hits the reverse fault at 224 m, turns again to a 45° deviation angle and passes through the shale/oil interface, where two sharp variations corresponding to two interfaces are captured.

3.2.4 Summary

In summary, this section presents a fast and memory-efficient solver for the LWD application. A flipped TF/SF DGFD method is proposed to avoid both the excessive meshes around the source and the elementwise integration of the scattering objects for the moving tool. The new method is then demonstrated to honor the invariance of the system matrix, and therefore can handle multiple RHSs all at once with PARDISO. A cylindrical lithologic reservoir model is discretized by tetrahedra/bricks mixed meshes to validate our solver. A large-scale oil trap model including a salt dome, a fold and a fault is presented with tetrahedra/hexahedra mixed meshes and curved domain decompositions to show that our memory-efficient solver can save up to 70% unknowns, which increases the computing capability for the LWD application under the same computing resource greatly. For long wellbore trajectories of thousands meters, it is reasonable to break the whole trajectory into independent small problems; the flipped TF/SF DGFD method can then help to enlarge the computational domain for each small problem, and correspondingly reduce the number of independent small problems.

4

Hydraulic Fracture Reconstruction under Energized Casing through Approximate Casing Model and Deep Learning

Part of this chapter was originally published in [Zhang, R., Sun, Q., Zhang, X., Cui, L., Wu, Z., Chen, K., Wang, D., and Liu, Q. H. (2020), Imaging Hydraulic Fractures Under Energized Steel Casing by Convolutional Neural Networks, *IEEE Transactions on Geoscience and Remote Sensing*] and is reprinted with permission. I designed the algorithm, developed the code and wrote the article.

Imaging hydraulic fractures is of paramount importance to subsurface resource extraction, geologic storage, and hazardous waste disposal. The use of electrically conductive proppants and current energized steel casing provides a promising approach to monitor the distribution of fractures. In this chapter, a borehole-to-surface system is employed to energize the steel casing and measure electric and magnetic fields on the ground. A CNN is then trained to learn the relationship between the measured field pattern and the parametrized fracture, namely the lateral extent and direction. To accelerate the generation of training data with limited accuracy loss, an approximate hollow casing is modeled by the ITBC with ten-fold magnified radius and reduced conductivity. Two training strategies are then presented with grid search of the network's hyperparameters. The well-trained CNN shows good generalization to unseen fracture conductivity, the true casing model, as well as white Gaussian noise. We also apply the CNN to image irregular fractures and obtain reliable results even under strong noise, indicating a promising imaging technique for more complicated fractures. Finally, we introduce the deep transfer learning to the full wave inversion problem for the first time and further improve the CNN's accuracy and efficiency.

4.1 Steel Casing

With the economic benefit reduction of the petroleum industry, how to maintain and exploit the available resources becomes more important, for example, the cased borehole. The steel casing extending from the surface to the deep gas layer strongly influences the EM field in borehole-to-surface configurations, attenuating the signal radiated from tools significantly and then masking the useful information. Therefore, the EM technique is not available under the presence of casing until around 1990 [Augustin et al. (1989)]. Currently, one promising application is to use the existing conductive infrastructure as extended current transmitters to monitor the hydraulic fracture or CO_2 reservoir, amplifying the signal without installing new equipment [Ahmadian et al. (2018); Puzyrev et al. (2017)]. Both analytical [Cuevas (2014)] and numerical work is ongoing to guide the related field tests. One numerical way is to model the casing as a conductive hollow to boost the source away from the casing [Commer et al. (2015); Um et al. (2015)], just like through casing tool modeling [Fang et al. (2017)], where the source doesn't touch the casing. Some simplification of the hollow casing is proposed to relieve the computational burden [Um et al. (2015); Puzyrev et al. (2017)] by using solid rectangular prism or solid cylinder. Although the DoF can be saved to some degree, the flexibility to model the casing of varying geometry, thickness and conductivity can be reduced, which is very important when considering corrosion effects [Martin et al. (2017)].

4.2 Hydraulic Fracturing

Hydraulic fracturing is widely used to enhance hydrocarbon production from sand and shale formations. Although this technology has advanced rapidly in recent years, reliable diagnoses and appraisal of various fracture properties (dimensions, orientation, conductivity, and proppant placement) suffer from excessive measurement uncertainties. These properties are critical to predict stimulated reservoir volume, which can be further applied to optimize completion strategies, improve the efficiency of resource usage, and predict the reservoir productivity [Hoversten et al. (2015)]. Various technologies for fracture diagnosis are commercially available [King et al. (2010); Cipolla et al. (2000)]. The predominant method of hydraulic fracture geometry inference is the microseismic technology [Maxwell (2014)]. The mapping of microseismicity associated with pumping or injection activities and the inversion for hypocenter location may provide an indication of the extent and dominant direction of newly generated fractures from the stimulated wellbore [Aarre et al. (2012)]; however, the relationship between hypocenter location and fracture genesis is not necessarily straightforward [Tafti et al. (2013)] and can be confounded by low impedance flow paths along existing fractures and reactivation of previously unmapped faults [Rutledge and Phillips (2003)]; further, the fracture is hard to detect after pumping or injection activities.

To provide adequate resolution and precision for determining the extent of hydraulic fractures, an alternative method is to use EM geophysical sensing, as widely used to map economic mineral deposits and fluid-bearing properties of reservoir rocks over the past eight decades. The current generation of deep EM tools [Zhang et al.

(2019b)] is operated by interpreting intrinsic differences in the electrical conductivity of hydrocarbon-bearing layers and the rest of the formation [Løseth and Ursin (2007)]. Additionally, the difference between the conductivity of injected fluids and that of connate fluids may be large enough to be exploited by deep EM tools. To increase the ability to sense the spatial extent of the injected materials for localization of proppants and fluids in the stimulated reservoir environment, contrast-agent additives or proppants with enhanced EM properties are injected into the fracture in the latest research [Ahmadian et al. (2018); Palisch et al. (2018)]. To amplify the scattered fields from the fracture as well as keep the whole experimental setup practical, the casing is used as an energized antenna to stimulate the fracture and receiver arrays are usually placed on the ground[Weiss et al. (2016); Puzyrev et al. (2017)].

The inversion of the hydraulic fracture, like many other 3D geophysics problems, can be divided into voxel-based inversion and model-based inversion. The voxel-based inversion is more general as all voxels can be tuned to recover the realistic target, which therefore, generates a large number of unknowns to recover. The local optimization is then preferred due to its cheaper computational burden compared to the global optimization [Nocedal and Wright (2006)]. However, for the highly non-linear inverse problem like the hydraulic fracture inversion, the reconstructed result of the local optimization usually differ from the true target and highly dependent on the initial guess. Moreover, the fracture is fundamentally a 2D structure and only a low-resolution bulk image can be obtained by the voxel-based inversion in the recent work [Palisch et al. (2018)]. On the contrary, the model-based inversion restrains the target under specific models and parametrizes the problem, which reduces the unknowns greatly while keeping the high resolution of images. However, no matter voxel-based or model-based inversions need to iteratively call nontrivial forward modeling solvers and are time-consuming, making it only feasible to provide a final estimation when the hydraulic fracturing is completed and not feasible to monitor the fracture's growth in real time. Fortunately, the rapid development of ML in recent years provide a potential solution to greatly accelerate the inversion. In ML-based inversion, forward modelings are only involved in data preparation and highly parallelizable; when reconstructing the fracture, machine learned mappings are evaluated instead of expensive forward modelings; therefore, it is promising to monitor the growth of the fracture in real time and give the fracturing operators instant feedback. The current advances in ML-based inversion mainly apply the CNN to the full wave inversion, focusing on either developing the network architectures [Wei and Chen (2018)] or applying to different scenarios [Puzyrev (2019)]. The hydraulic fracture imaging under energizing casing, however, is different from other applications due to its 2D geometric property as well as highly multiscale scenarios; this problem is not solved by the ML-based inversion method yet. To fill this gap, a CNN-based inversion method is proposed in this chapter to image the fracture by estimating its lateral extent and direction. An approximate forward model for the hollow casing is proposed with the ITBC as well as magnified radius to accelerate the data generation. Hyperparameters of the CNN are determined by grid search with two different training strategies. The well-trained CNN generalizes well to the true casing model and unseen conductivity. A composite fracture is finally reconstructed to show the CNN's reliable performance under strong noise.

4.3 Forward Model for Hydraulic Fractures Imaging under Energized Steel Casing

4.3.1 Model Specification

A simplified well model is employed in this chapter to investigate the hydraulic fracture imaging under casing energization as shown in Fig. 4.1. The vertical part of the well is 1.5 km deep, the horizontal part is 2 km long, and the connection part

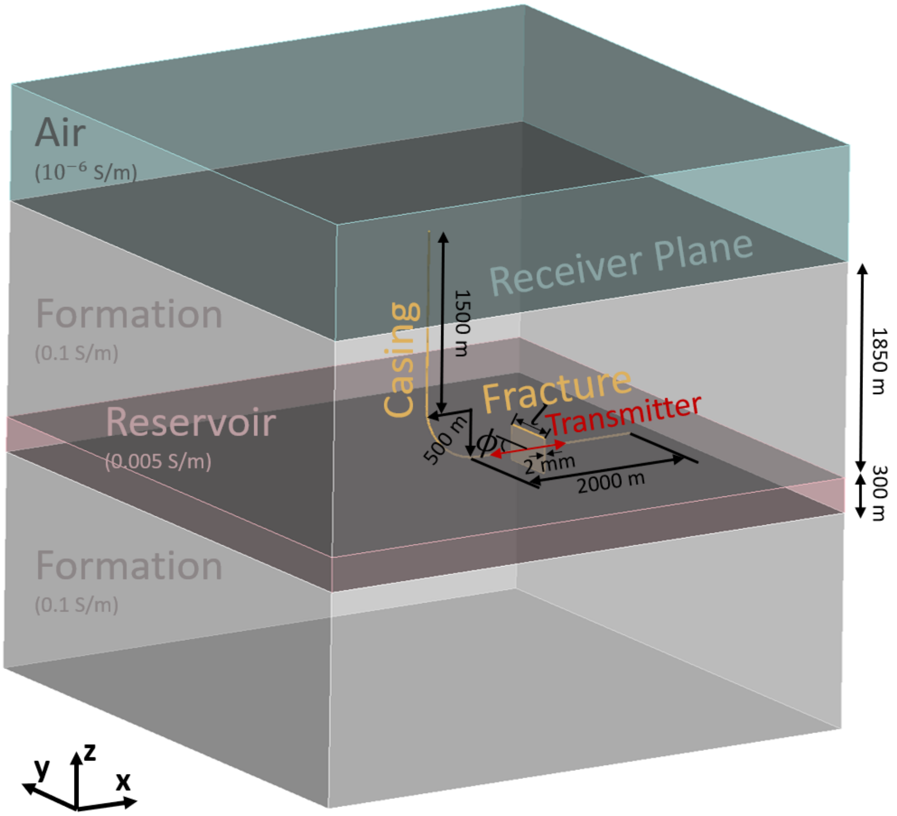


FIGURE 4.1: Diagram of the hydraulic fracture under energized casing

is one quarter of torus with 500-m radius. The conductivity (relative permeability) of casing is 2.5×10^6 S/m (50), and its diameter (wall thickness) is set to 0.3 m (6 mm). The horizontal well lies in the middle of a 300-m-thick reservoir layer of 0.005 S/m, while both top and bottom layers are supposed to be conductive (0.1 S/m). The air layer is added by a small conductivity (10^{-6} S/m) to avoid the extremely singular finite element matrix, and higher values of air conductivity were found to cause small errors in measurements on the ground [Puzyrev et al. (2017)].

The fracking fluid is injected into the well to create cracks in the reservoir layer; due to the surrounding pressure, the fracture usually propagates in the vertical direction and then bounded within the overburden and underburden layers (300 m here) that have higher minimum horizontal stress and/or higher strength than those of the reservoir. The horizontal extent of fracture is non-determinant and usually hundreds of meters long, with an arbitrary azimuthal angle between the horizontal well. The injected electrically conductive contrast-agent additive can hold the fracture open around 2 mm with conductivity up to 500 S/m. Since the fracture length together with azimuthal angle can indicate the general trend of the fracture, they are among the most important parameters in fracture imaging and consequently set as the target parameters in our model-based inversion. To simplify the fracture modeling in the following, we suppose the fracture is always rectangular with a tunable length l, azimuthal angle ϕ (the angle between the negative x direction and the top edge of fracture in the positive y zone) and conductivity σ, while the height and thickness are fixed at 300 m and 2 mm. Note that σ may vary in real cases but not as important as l and ϕ in fracture imaging; therefore it only serves as a variable to increase the model's generality, but not a target parameter.

A borehole-to-surface system is employed to energize the casing and measure the fields on the ground. A 500-m-long line source of 40 A is embedded at the horizontal part of the casing to inject current, with positive and return electrodes at opposite sides of the fractures; the working frequency is selected as 16 Hz, corresponding to five skin depths between fracture and ground. Although higher frequency can provide better resolution, the stronger dissipation will decrease the scattered fields on the ground to below 10^{-9} V/m (A/m) and to approach the noise floor [Tietze et al. (2015)]. 64×64 uniformly-spaced receivers are placed around the casing on the ground to measure two tangential electric field components and all three magnetic field components [Puzyrev et al. (2017)].

4.3.2 Approximated Hollow Model for the Casing

The modeling of the steel casing effect is challenging since the radius of the casing is small in comparison with its vertical dimensions and the thickness of the casing wall is even smaller. The combination of the numerical matching method and integral methods is explored to model the casing effect [Fang et al. (2019)]. More studies employ the finite-difference and finite-elements methods and show the feasibility to include the realistic casing geometries as a hollow cylinder; however, the direct discretization of a few kilometers long hollow cased wells requires around one hundred million unknowns; to reduce the unknowns and make running a plurality of forward modelings practical, some approximations of the casing are proposed. One approximation is to model the steel casing as a solid prism, whose conductivity is determined by keeping the cross-sectional conductance of the prism the same as the hollow well [Commer et al. (2015); Um et al. (2015)]. The other is to keep the hollow model for the casing, but replace the tiny volumetric meshes for the thin casing wall with surface meshes of ITBC [Sun et al. (2017)]. Here, we follow the second spirit, but further reduce the unknowns by amplifying the radius of the casing ten-fold and reducing the conductivity to about one-tenth; consequently, the smallest meshes jump from centimeters to decimeters, reducing up to nine-tenth unknowns and making one forward modeling of the kilometers long casing less than ten minutes.

The use of ITBC to model the casing is already verified in [Sun et al. (2017)], and here we dive deeper into the physics of the modeling. As other boundary conditions, the ITBC also specifies special relationships between tangential electric fields and tangential magnetic fields on the boundary. However, since ITBC belongs to internal boundary conditions, fields exist on both sides of the boundary. The fields at different sides of the casing wall are linked by the ITBC as

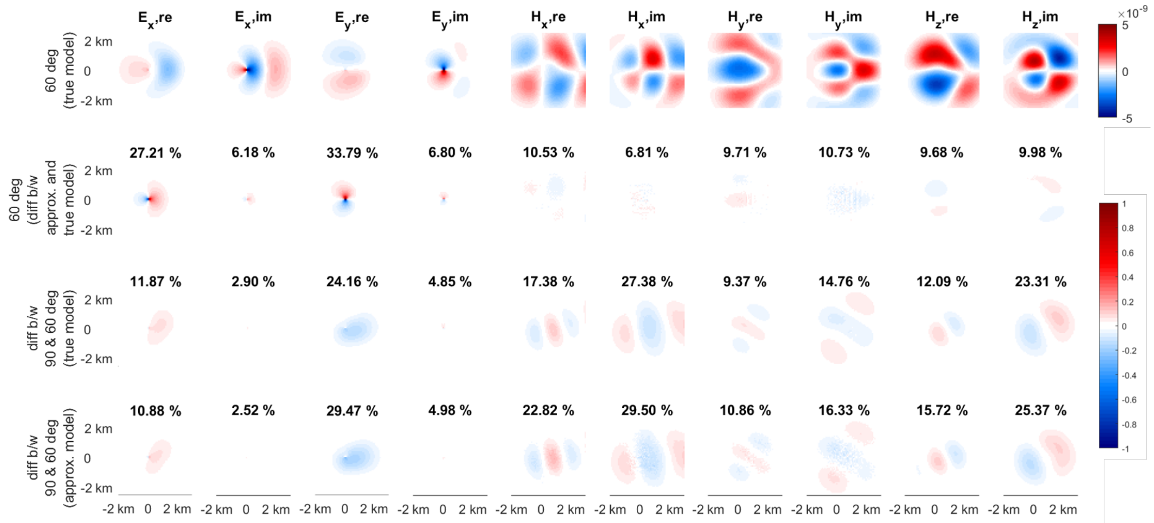


FIGURE 4.2: First row: scattered electric (magnetic) fields ($\times10^{-8}$ V/m (A/m)) from a hydraulic fracture of $l = 500$ m, $\phi = 60°, \sigma = 100$ S/m using the true casing model; Second row: normalized difference of scattered fields by approximated casing model and true casing model with NRMS difference; Third (Fourth) row: normalized difference between fractures of $\phi = 90°$ and $\phi = 60°$ under true (approximated) casing model with NRMS difference.

$$\hat{n} \times (\mathbf{H}^+ - \mathbf{H}^-) = -i\omega\varepsilon\kappa\hat{n} \times ((\mathbf{E}^+ + \mathbf{E}^-) \times \hat{n}) \tag{4.1}$$

$$\hat{n} \times (\mathbf{E}^+ - \mathbf{E}^-) = i\omega\mu\kappa\hat{n} \times ((\mathbf{H}^+ + \mathbf{H}^-) \times \hat{n}) \tag{4.2}$$

where

$$\kappa = \frac{e^{ik_cd} + e^{-ik_cd} - 2}{ik_c(e^{ik_cd} - e^{-ik_cd})} = \frac{1}{k_c}\tan\left(\frac{k_cd}{2}\right), \tag{4.3}$$

$\hat{n}$ is the normal to the casing wall of wave number k_c and thickness d, and + and − refer to the outer and inner surfaces of the casing wall. As shown above, the ITBC is a simplified 1D model for wave propagating vertical to the thin layer and fields polarizing parallel to the thin layer, which is exactly the case for energized casing: the electric fields inside the casing walls are dominated by induced currents flowing along the longitudinal direction and decaying exponentially in the radial direction due to the high conductivity of the casing; the primary (secondary) magnetic fields are azimuthally surrounding the line source (induced current on the casing). Note that ITBC requires the radius of the curvature of layer much larger than both the skin depth and thickness of the layer, which is satisfied in our case as the casing radius is 150 mm and the casing skin depth (thickness) is only 10 mm (6 mm), making the layer locally planar. As a first-order approximation, we can view the total number of meshes as a bilinear function of the mesh number at the casing's cross section and the casing's length, where the former one is reciprocal to the minimal size of meshes. Since the ITBC does not discretize the interior zone of the casing wall, extremely tiny meshes comparable to the thickness of the casing wall will expand about 20 times to approach the radius of the casing and save up to 19/20 unknowns.

However, even with the help of the ITBC, kilometers of energized casing can still produce tens of millions of unknowns to solve, which may take around one hour in a normal 12-core CPU and more than 100 GB memory with the state-of-the-arts of iterative solver (FGMRES). As mentioned above, the current minimal meshes are governed by the casing radius; therefore, to further reduce up to 9/10 of the unknowns, we amplify its radius to ten-fold. To neutralize the increased conductance, we decrease the conductivity of the casing to its 1/10. Although the conductance of the casing is then invariant, its contact area to the conductive formation is enlarged, which leaks more induced currents into the formation than the real case. To compensate for the larger leakage, the conductivity of the casing needs to increase to guide the currents back to the casing. This ratio, however, is hard to be determined analytically and here we use a trial and error method to obtain the ratio as 1.35. The same case with the amplified casing will then only require around 2 million unknowns, 10 GB memory and 6 minutes to solve, which tremendously reduces the computational burden and makes the generation of a large amount of training/validation data available.

4.3.3 Approximation Error and Sensitivity Analysis

To test the accuracy of the approximated casing model, we place a 500-m-long 100-S/m-conductive hydraulic fracture 750 m away from the vertical borehole and rotate it to $\phi = 60^\circ$. The scattered fields from hydraulic fractures under the true casing model is shown in the first row of Fig. 4.2 and the normalized difference of scattered fields between approximated and true model is shown in the second row. Note that the total electric and magnetic fields have similar amplitude in this case and therefore their absolute scattered fields are plotted together without normalization, which reveals the exponential decay of the electric fields and near-constant magnetic fields away from the casing. Since our approximation tries to preserve the strength of the induced current along the casing, the magnetic fields on the ground are also preserved according to the Ampere's law $\nabla \times \mathbf{H} = \mathbf{J}$, which is not sensitive to the specific current flow path. However, the real part of the electric field is directly coupled from the currents according to the continuity of the currents and then sensitive to the specific current flow path. Consequently, from the second row we can find that the approximated model can reproduce similar scattering patterns as the true model, including the imaginary parts of electric fields and the complex magnetic fields; the normalized root mean square (NRMS) error for these eight components are generally between 6% and 11%. For the real parts of the electric fields, the enlarged casing radius reverts the fields phase near the casing and introduces much larger errors (around 30%). The third (fourth) row shows the normalized difference of scattered fields between fractures of $\phi = 90^\circ$ and 60° under the true (approximated) casing model; the similar field differences indicate that the approximated model can capture the fracture variation well; however, the difference of imaginary electric fields is too small (less than 5% NRMS) to distinguish the fracture orientation, while the NRMS of magnetic fields can reach up to 30%. The low discriminability of electric fields

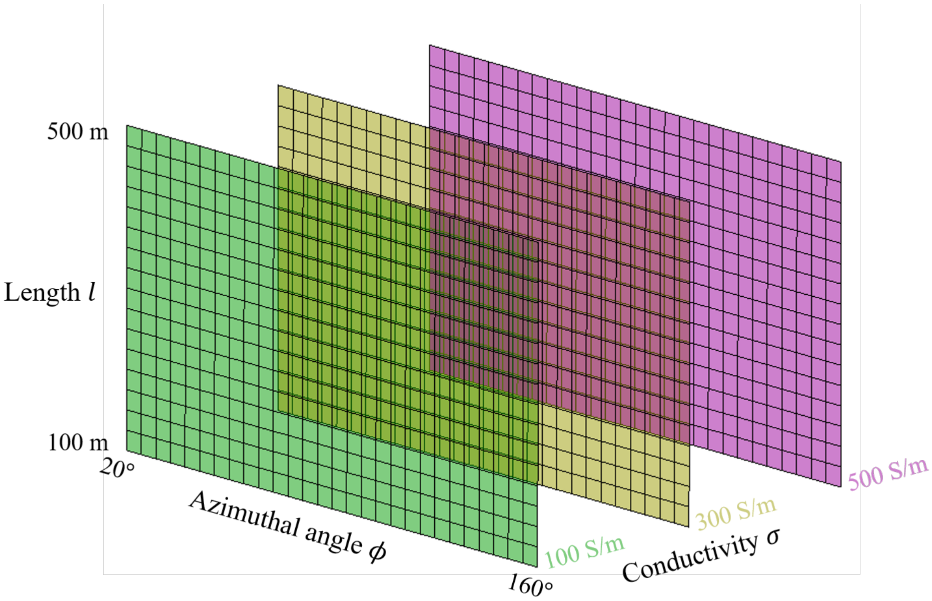


FIGURE 4.3: Diagram of the training/validation data space. The azimuthal angle ϕ and length l are dicretized densely to guarantee the inversion accuracy; the conductivity σ is discretized coarsely to reduce the size of the dataset while improve the generalization ability to conductivity. Each point of intersection represents one sample.

can be attributed to its fast dissipation in the high conductive formation, while the magnetic field decouples with the electric field at low frequency and therefore can preserve more information. Considering the inaccurate approximation of the real electric field and the low discriminability of the imaginary electric field, only the magnetic field is used in the following inversion.

4.4 Convolutional Neural Networks

4.4.1 Data Preparation

The target parameters' domains are set as [20°, 160°] and [100 m, 500 m] respectively for azimuthal angle ϕ and length l of the fracture, which cover the most possible scenarios fractures may end up with. To guarantee a good accuracy for the inversion, we discretize the domain densely with a step size of 5° (25 m) and corresponding relative step size 4% (6%) for ϕ (l), as indicated by surface meshes in Fig. 4.3. When fixing all other parameters, the structured dataset will include 493 samples. However, these samples may overfit the model to a specific fracture conductivity, which is not expected since the actual conductivity of fracture can vary a lot due to the surrounding environment. To boost the generality of the model for different fracture conductivities, we triple the dataset by employing the above target parameter discretization under three different conductivities 100 S/m, 300 S/m and 500 S/m as shown by three surfaces in Fig. 4.3; the size of dataset consequently increases to 1,479. Note that for fracture pairs of complementary azimuthal angles, their scattered fields' components on the ground are either symmetric or anti-symmetric to each other, with regard to the horizontal borehole's vertical projection to the ground; therefore, we can just compute half of the dataset and obtain the other half trivially by flipping the corresponding dataset, termed as data augmentation in ML fields. We then can generate the structured dataset by running the forward modeling approximately $17 \times (28+2) \times (3+1)/2 = 1,020$ times, where the extra +1 represents the computation of the incident field for each fracture geometry. Theoretically, the incident field is the same independent of the shape of fractures; however, subtracting the incident field from the total field of different meshes will introduce some noise due to the mesh inconsistency; therefore, it is a good practice to employ the same mesh to compute total/incident fields for the same fracture. The corresponding runtime is around 100 hours for the approximated casing model and 1,000 hours for the true casing model, both on a single computer of 12 cores.

4.4.2 Neural Networks Architectures

CNN is one specialized kind of networks for data that has a grid-like topology, which has played an important role in DL [Goodfellow et al. (2016)] and led to the recent breakthroughs in image classification [Krizhevsky et al. (2012); Simonyan and

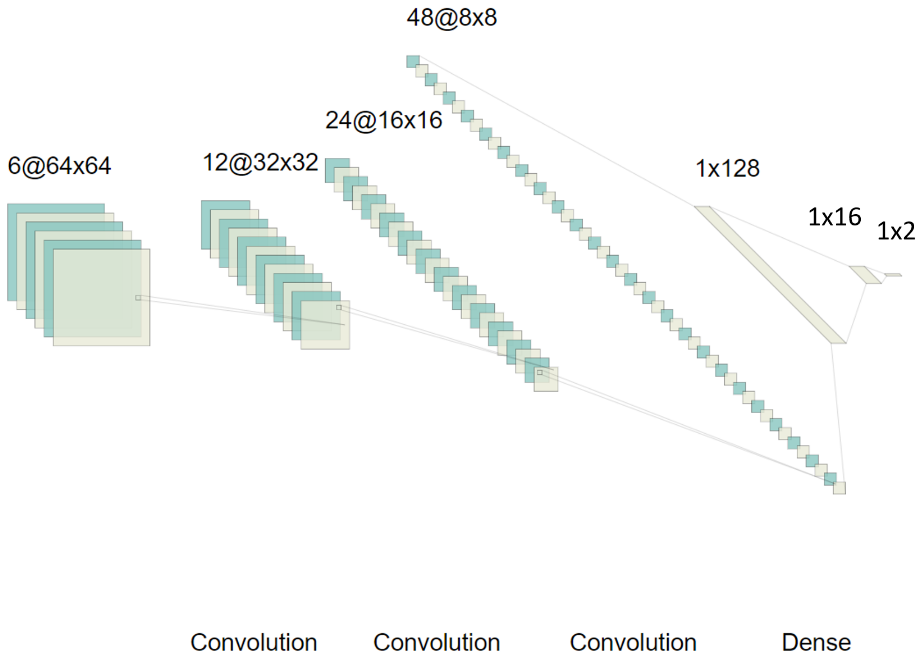


FIGURE 4.4: Diagram prototype of CNN networks for the fracture imaging. The input data is 64×64 scattered fields with six channels; four convolutional layers with input of the format "channel number @ pixel number (width) × pixel number (height)" are connected to two fully connected layers (FCLs) with output node numbers 128, 16 and a output layer.

Zisserman (2014); He et al. (2016)], and some other fields like video analysis and time-series data processing. As indicated by its name, a convolutional network employs a convolution operation to low-pass filter the input data, which replaces the general matrix multiplication in fully connected networks and therefore significantly reduces the number of free parameters and makes stacking a large amount of layers feasible. The computational burden is further reduced by pooling and downsampling the output of convolution operations, which can also make the output data approximately invariant to small translations of the input. Normally, the max pooling operation that takes the maximum within a rectangular neighbourhood is employed.

A typical architecture of a convolutional network consists of a set of layers, each of which contains several filters to extract various features in the input. The most common CNN architectures have been developed for single or multiple label classifications (discrete label) and regressions (continuous label) with a fully connected layer at the end [Krizhevsky et al. (2012); Simonyan and Zisserman (2014); He et al. (2016); Zhang et al. (2018)]. Since fracture imaging has continuous azimuthal angle and length of the fracture as target parameters, it is then a regression problem. The prototype of the network we choose is shown in Fig. 4.4; the dimensions of input data are $64 \times 64 \times 6$, where 64×64 grid-like receivers record six-channel signals including real and imaginary parts of three scattered magnetic field components. The data is then filtered by several convolutional layers; each convolutional layer will start from a classical 2D convolutional kernel of $3 \times 3 \times N_{channel}$, where the channel number $N_{channel}$ starts from 6 and doubles within every convolutional layer; the stride length of all convolutions is fixed at one, because larger stride shrinks the image dimensions fast to reduce the complexity and speedup the training process of the deep neural network but lose some accuracy, which is more suitable for very deep neural networks to preprocess high-resolution images; zero padding is employed to keep the height and width of outputs the same as those of inputs, which helps to make the height and width of every convolutional layer as the power of two and therefore facilitate the process of cascading different convolutional layers; moreover, from Fig. 4.2 we can observe that scattered fields at boundary are generally much smaller than those at center, which conforms to the diffusion property of EM waves in lossy media; therefore, zero padding will not induce obvious jumps at boundary and cause the smear effect. The outputs of convolution are then activated by a rectified linear unit (ReLU) function, which can reduce the probability of vanishing gradient and help converge faster. The batch normalization is then applied to reduce the covariance shift and support higher learning rates, because activation outputs are guaranteed in a certain range. Subsequently, the data is downsampled by a 2×2 max pooling layer to ensure the invariance to small input translations as well as reduce the computational burden. Finally, a dropout layer is added to ignore part (r percent) of outputs from previous layer randomly to imitate a layer with a different number of nodes and connectivity to the prior layer, thus reducing overfitting. Although the dropout layer typically requires $\frac{1}{1-r}$ times larger networks and more iterations than the networks without dropout layer, its effectiveness and efficiency still make it one of the most common regularization techniques in DL, especially when the dataset is not large, because the dropout can also be viewed as a process of constructing new inputs multiplied by noise. Moreover, the dropout layer only requires a small dropout rate with the help of batch normalization, thus only increasing the networks a little. After several convolutional layers, a couple of fully connected layers (FCLs) are employed to extract the estimation of two target parameters. The FCLs have similar structure as the convolutional layer, with a substitution of the convolution operation to general matrix multiplication and an elimination of max pooling. Additionally, the dropout is also not preferred when the number of outputs is not large enough.

4.4.3 Training and Testing Design

Since the generation of synthetic data under true casing model is time consuming and not feasible to generate even a small dataset of 1,428 samples, the training/validation data are then prepared by the approximated casing model instead; the specific trained CNN consequently fits the approximated casing model. To test the performance of the CNN, we need to prepare two sets of testing data, namely the approximated testing data and the true testing data; the former is obtained by randomly selecting 5% of the 1,428 samples under approximated casing model to evaluate the generalization ability of the network within the same casing model; the latter is from the true casing model and used to evaluate the performance of the network on more realistic data, including 75 samples from fractures of ϕ = 30, 60, 90, 120, 150 degree, l =100, 300, 500 m and σ = 100, 200, 300, 400, 500 S/m. Note that we down sample ϕ and l to save time but increase the density of σ to test the corresponding generalization ability.

To take full advantage of limited dataset and improve the CNN's robustness to different input fields, we use a 5-fold cross validation by splitting the rest 1,355 samples into five groups and taking each group as validation data successively and other four as training data; consequently, we will obtain five independent sets of weights. When evaluating the CNN's performance, we apply the CNN quintuple to testing data separately and the mean of obtained five estimations will be the final result, while their standard deviation can serve as a proxy for the consistency.

Both mean absolute error (MAE) (4.4) and mean squared error (MSE) (4.5)

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |\phi_i^{np} - \phi_i^{nt}| + |l_i^{np} - l_i^{nt}|}{N} \tag{4.4}$$

$$\mathrm{MSE} = \frac{\sum_{i=1}^{N} |\phi_i^{np} - \phi_i^{nt}|^2 + |l_i^{np} - l_i^{nt}|^2}{N} \tag{4.5}$$

are popular loss functions and tested in our model, where superscripts np (nt) represent normalized predicted (normalized true) value and N is the number of training or validation or testing data. The normalization is a linear mapping from either ϕ or l to a [0,1] interval. Numerical experiments show that MAE can always achieve better convergence and testing errors in our model; therefore we employ MAE in the following discussion.

4.4.4 Grid Search for the Hyperparameter of the CNN

To achieve the balance between overfitting and underfitting given the training / validation / testing data, we need to tune several hyperparameters that have not been determined in Section 4.4.2, which includes the number of the convolutional layers, the number of cascaded convolutional kernels inside each convolutional layer, the position and dropout rate of the dropout layer, the width and depth of the FCL, and so on. To perform the grid search of these parameters efficiently, we further fix several of them as:

1. Use two FCLs to connect the convolutional layers and the output layer.

2. Place the dropout layer after every convolutional layer but not after two FCLs, because the dropout layer will probabilistically reduce the network capacity and work better with a large number of output nodes.

We then run a grid search for the following four parameters sequentially, which means only one parameter is swept over at each time while the other three fixed; the initial setup is five convolutional layers with the last feature map of $2 \times 2 \times 192$, dropout rate of 0.25 and two FCLs of 128 and 16 output nodes; after sweeping one parameter, its value will be updated from the initial one to the optimized one for the following grid search:

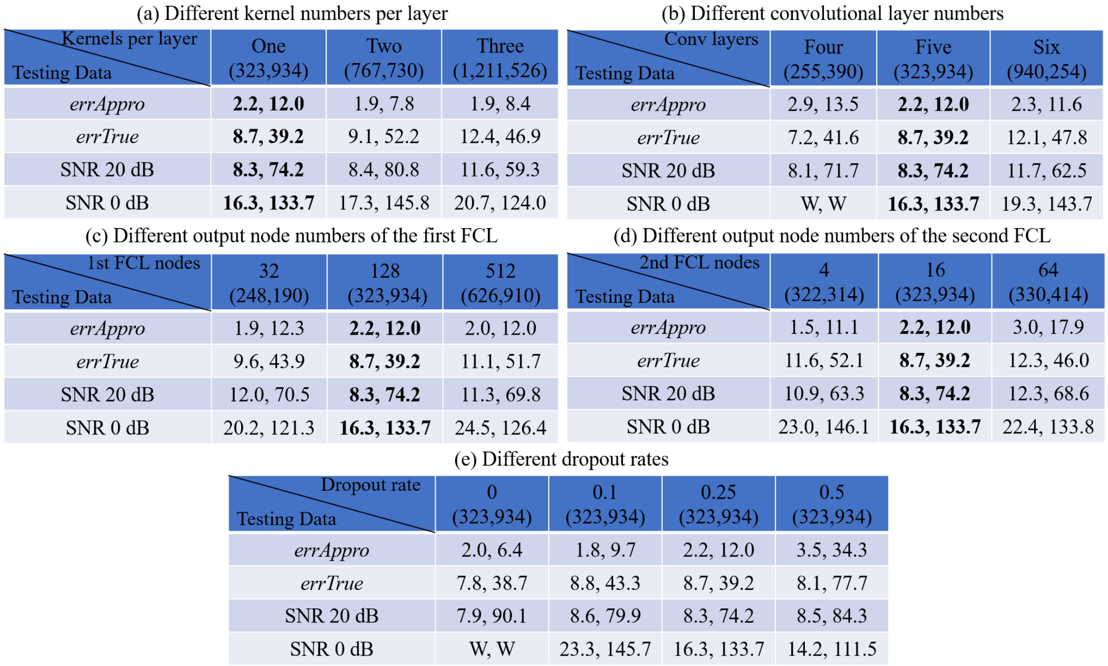
1. We first grid search the cascaded convolutional kernel number per layer over one, two and three, because it can dramatically change the number of convolution operations (complexity) of the CNN.

2. The convolutional layer number is then swept over from four to six, with a last feature map of either $4 \times 4 \times 96$ or $2 \times 2 \times 192$ or $1 \times 1 \times 384$; this wraps up the configuring of the convolutional layers.

3. Consequently, the configuration of FCLs is tuned by sweeping two output dimensions separately; the candidate of the first (second) FCL is 32, 128 and 512 (4, 16 and 64).

4. Finally, we sweep the dropout rate over 0, 0.1, 0.25 and 0.5 to accommodate different input SNR.

Table 4.1: Testing error in ϕ (°), l (m) pair by pure approximated training data



(a) Different kernel numbers per layer

Kernels per layer Testing Data	One (323,934)	Two (767,730)	Three (1,211,526)
errAppro	**2.2, 12.0**	1.9, 7.8	1.9, 8.4
errTrue	**8.7, 39.2**	9.1, 52.2	12.4, 46.9
SNR 20 dB	**8.3, 74.2**	8.4, 80.8	11.6, 59.3
SNR 0 dB	**16.3, 133.7**	17.3, 145.8	20.7, 124.0

(b) Different convolutional layer numbers

Conv layers Testing Data	Four (255,390)	Five (323,934)	Six (940,254)
errAppro	2.9, 13.5	**2.2, 12.0**	2.3, 11.6
errTrue	7.2, 41.6	**8.7, 39.2**	12.1, 47.8
SNR 20 dB	8.1, 71.7	**8.3, 74.2**	11.7, 62.5
SNR 0 dB	W, W	**16.3, 133.7**	19.3, 143.7

(c) Different output node numbers of the first FCL

1st FCL nodes Testing Data	32 (248,190)	128 (323,934)	512 (626,910)
errAppro	1.9, 12.3	**2.2, 12.0**	2.0, 12.0
errTrue	9.6, 43.9	**8.7, 39.2**	11.1, 51.7
SNR 20 dB	12.0, 70.5	**8.3, 74.2**	11.3, 69.8
SNR 0 dB	20.2, 121.3	**16.3, 133.7**	24.5, 126.4

(d) Different output node numbers of the second FCL

2nd FCL nodes Testing Data	4 (322,314)	16 (323,934)	64 (330,414)
errAppro	1.5, 11.1	**2.2, 12.0**	3.0, 17.9
errTrue	11.6, 52.1	**8.7, 39.2**	12.3, 46.0
SNR 20 dB	10.9, 63.3	**8.3, 74.2**	12.3, 68.6
SNR 0 dB	23.0, 146.1	**16.3, 133.7**	22.4, 133.8

(e) Different dropout rates

Dropout rate Testing Data	0 (323,934)	0.1 (323,934)	0.25 (323,934)	0.5 (323,934)
errAppro	2.0, 6.4	1.8, 9.7	2.2, 12.0	3.5, 34.3
errTrue	7.8, 38.7	8.8, 43.3	8.7, 39.2	8.1, 77.7
SNR 20 dB	7.9, 90.1	8.6, 79.9	8.3, 74.2	8.5, 84.3
SNR 0 dB	W, W	23.3, 145.7	16.3, 133.7	14.2, 111.5

To adapt the number of training epochs to different networks to avoid overfitting, early stopping is employed to stop the optimization when the validation error does not decrease after 200 epochs; small batch size (16) is employed to secure good generalization ability for CNNs, because large-batch methods tend to converge to sharp minimizers of the training function [Keskar et al. (2016)]; the adaptive learning rate optimizer Adam is used due to its computational efficiency and fast convergence, and the learning rate is chosen as 0.001. The training is run on one Tesla K80 GPU from the Google Colab, which normally takes 500 to 1200 epochs to converge; the runtime per epoch largely depends on the network's parameter number and at most 8 s in the following tests. Table 4.4 shows the averaged testing errors $errAppro$ ($errTrue$) for the approximated (true) testing data with the format as ϕ (°), l (m) pair. Additionally, white Gaussian noise is added to all six channels of the true casing data, with signal-to-noise ratios (SNR) of 20 dB and 0 dB; since different channels have different root mean square values, their noise should have different variances to secure the same SNR. Note that in Table 4.4: 1) the number inside the parenthesis is the total parameter number of the corresponding network; 2) errors are marked as "W" (short for wrong) when they are larger than 30° or 160 m; 3) the column corresponding to the smallest testing errors/optimal parameter is marked bold. From Table 4.4 we can observe that,

1. Table 4.4 (a) shows that *errAppro* from CNNs with two or three cascaded kernels per layer are smaller than that with one kernel per layer. However, the CNN with one kernel per layer has the smallest $errTrue$, indicating CNNs with more kernels cascaded per layer overfit to the approximated data and generalize worse than the CNN with one kernel per layer, by viewing the data from the true casing model as specific noise pattern polluted data from the approximated casing model.

2. Table 4.4 (b) shows the impact from the number of convolutional layers, where the CNN of four convolutional layers malfunctions with 0 dB error and the CNN of six convolutional layers generally performs a little worse than the five convolutional layers.

3. Table 4.4 (c) and (d) tune the output nodes of different FCLs. This manipulation shows much smaller influence to the testing errors than the above two subtables. The optimal results show a quasi-geometric relationship from the convolutional layers' output to the network's final output: 768 ⇒ 128 ⇒ 16 ⇒ 2.

4. Table 4.4 (e) shows that lower dropout rates typically have smaller *errAppro*, while the $errTrue$, especially with strong noise tends to favor large dropout rate. Here, we can view *errAppro* as errors for clean training data and $errTrue$ as errors for noise polluted data. The more serious the pollution is, the larger

dropout rate is required to increase the tolerance to noise.

5. The CNN of five convolutional layers, one kernel per layer and 128/16 FCL output nodes outperforms others; we can tune its dropout rate to adapt to different noise level: 0 for clean data, 0.25 for 20 dB noise and 0.5 for 0 dB noise.

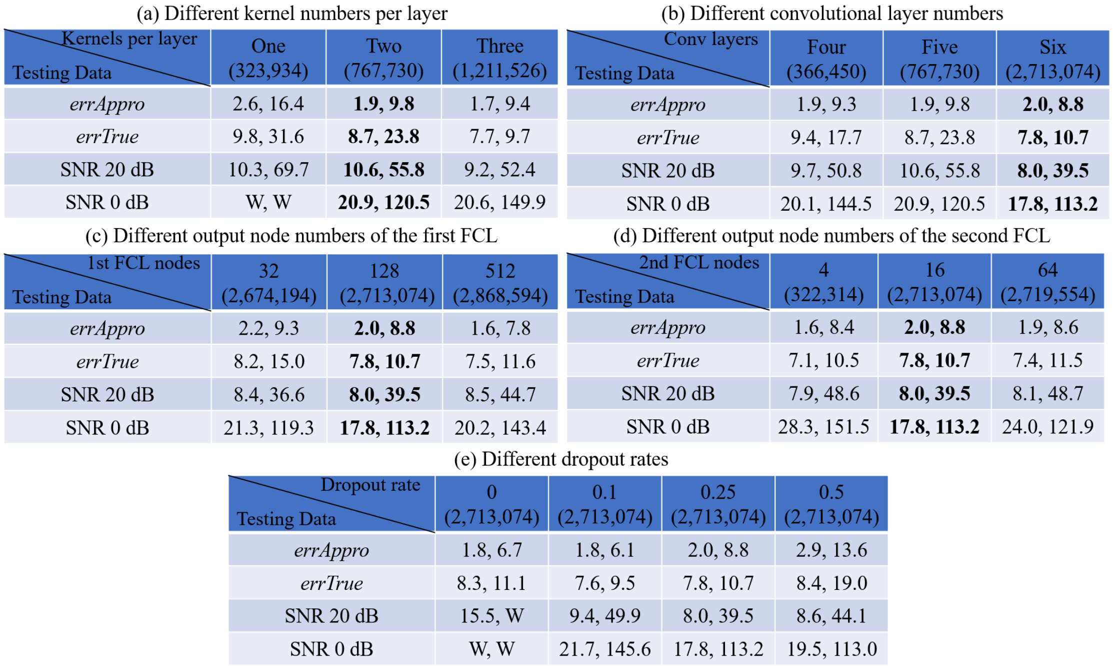
6. The estimation of ϕ is more robust to noise than that of l; for example, 20 dB noise does not normally increase the error for ϕ, but almost double the error of l.

7. $errTrue$ for CNN is generally around 3 times larger than corresponding $errAppro$, which can be attributed to the difference of the casing modeling as well as more boundary testings for the true casing model. Note that although not given here, the unseen $\sigma = 200$ S/m and 400 S/m will not introduce extra errors, indicating good generalization to σ.

4.4.5 Improve the CNN by Incorporating Small Number of True Model Data into Training

Table 4.4 shows a large accuracy loss from the approximated testing data to the true testing data, mainly because the CNN is trained by the approximated data, which differs from the true data by about 10%. To narrow the gap between these two testing errors, on the one hand, we can design more accurate approximated model; on the other hand, we can incorporate a small number of true data into the training process and guide the network to smoothly migrate to the true data. To make a fair comparison against the previous pure approximated data training strategy, we keep both the true and approximated testing data untouched, but add 54 true data for fractures of ϕ = 20, 45, 75, 105, 135, 160 degree, l =100, 300, 500 m and σ = 100, 300, 500 S/m. Note that all newly added data have counterparts in the

Table 4.2: Testing error in ϕ ($^\circ$), l (m) pair by incorporating true training data



(a) Different kernel numbers per layer

Kernels per layer Testing Data	One (323,934)	Two (767,730)	Three (1,211,526)
errAppro	2.6, 16.4	**1.9, 9.8**	1.7, 9.4
errTrue	9.8, 31.6	**8.7, 23.8**	7.7, 9.7
SNR 20 dB	10.3, 69.7	**10.6, 55.8**	9.2, 52.4
SNR 0 dB	W, W	**20.9, 120.5**	20.6, 149.9

(b) Different convolutional layer numbers

Conv layers Testing Data	Four (366,450)	Five (767,730)	Six (2,713,074)
errAppro	1.9, 9.3	1.9, 9.8	**2.0, 8.8**
errTrue	9.4, 17.7	8.7, 23.8	**7.8, 10.7**
SNR 20 dB	9.7, 50.8	10.6, 55.8	**8.0, 39.5**
SNR 0 dB	20.1, 144.5	20.9, 120.5	**17.8, 113.2**

(c) Different output node numbers of the first FCL

1st FCL nodes Testing Data	32 (2,674,194)	128 (2,713,074)	512 (2,868,594)
errAppro	2.2, 9.3	**2.0, 8.8**	1.6, 7.8
errTrue	8.2, 15.0	**7.8, 10.7**	7.5, 11.6
SNR 20 dB	8.4, 36.6	**8.0, 39.5**	8.5, 44.7
SNR 0 dB	21.3, 119.3	**17.8, 113.2**	20.2, 143.4

(d) Different output node numbers of the second FCL

2nd FCL nodes Testing Data	4 (322,314)	16 (2,713,074)	64 (2,719,554)
errAppro	1.6, 8.4	**2.0, 8.8**	1.9, 8.6
errTrue	7.1, 10.5	**7.8, 10.7**	7.4, 11.5
SNR 20 dB	7.9, 48.6	**8.0, 39.5**	8.1, 48.7
SNR 0 dB	28.3, 151.5	**17.8, 113.2**	24.0, 121.9

(e) Different dropout rates

Dropout rate Testing Data	0 (2,713,074)	0.1 (2,713,074)	0.25 (2,713,074)	0.5 (2,713,074)
errAppro	1.8, 6.7	1.8, 6.1	2.0, 8.8	2.9, 13.6
errTrue	8.3, 11.1	7.6, 9.5	7.8, 10.7	8.4, 19.0
SNR 20 dB	15.5, W	9.4, 49.9	8.0, 39.5	8.6, 44.1
SNR 0 dB	W, W	21.7, 145.6	17.8, 113.2	19.5, 113.0

approximated dataset; therefore, we do not increase the dataset in terms of ϕ, l, σ, but only expose true field patterns for a fraction of original approximated dataset to train the CNN. In terms of the 5-fold cross validation, the original approximated dataset and newly added true dataset are split independently to make sure similar amount of true data can be learned by the CNN quintuple. Further, we can view the new training strategy as data augmentation, where the time-consuming true data is augmented by a relatively large amount of approximated data to help generalize the model.

By applying the same optimization procedure, we can obtain Table 4.2, from where we can read that:

1. Table 4.2 (a) and (b) show that the configuration of six convolutional layers with two cascaded kernels per layer outperforms others; the increased complex-

ity of optimal architecture can be attributed to the more complex patterns it needs to learn to accommodate both the true and approximated data. Additionally, the added true data also prevent the larger networks from overfitting to the approximated data.

2. Compared to Table 4.4, *errAppro* has little change, but *errTrue*, especially l, is improved significantly to be comparable to *errAppro*, indicating the added true data successfully guide the CNN to fit the true testing data. With the increase of noise, the improvement becomes less significant, because the strong noise will dominate the error than the casing model difference.

3. The observations that large noise favors large dropout rate and ϕ is more robust to noise still hold.

Besides, the running time of the CNN quintuple for one set of unseen data is less than one minute, which is much faster than the traditional inversion, because the traditional inversion will include multiple forward modelings; even with the approximated model, it still requires more than one hour.

4.4.6 Comparison with KNN

To show the advantage of the well-trained CNN model, we choose the k-nearest neighbors algorithm (KNN) as a reference and tune its parameters to achieve the optimal performance (the number of neighbors is set as 2 and the weights are chosen as the inverse of their distances). When using the same training/testing data as the CNN in Section 4.4.5, we can get $errAppro = 1.7°, 17.3$ m and $errTrue = 15.9°, 117.8$ m. The *errAppro* is less than the step size, indicating that the simple KNN works well when only densely sampled ϕ and l require interpolation. The *errTrue*, however, has an angle (length) error larger than triple (fourfold) of the corresponding step size, comparable to the CNN's error under 0 dB noise. When examining the length

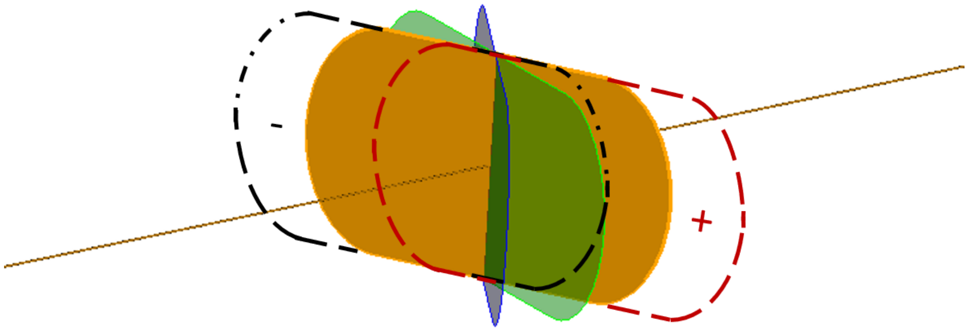


FIGURE 4.5: Diagram for nonrectangular fractures under three orientation angles ϕ = 60°, 90° and 120°; the 60° fracture is further translated 100 m along its two horizontal extension directions denoted by + and - dashed curves.

error more carefully, we can find that the error associated with seen σ = 100, 300, 500 S/m is 100.0 m and with unseen σ = 200, 400 S/m is 144.4 m. We can then conclude two reasons for the failure of the KNN: 1) The KNN's performance depends highly on the density of the dataset in terms of ϕ, l and σ; since we intentionally discretize the unimportant σ coarsely to shrink the dataset from 10,000 to 1,000, the unseen σ in testing data will introduce large errors. Conversely, the CNN is trained to be insensitive to various conductivity and therefore can perform much better on the testing set. 2) The KNN's straightforward interpolation strategy upon nearest neighbors cannot accommodate the difference between two casing models well, which introduces larger errors even on the data of seen σ than errors from the CNN, whose convolution operation narrows the gap between images of two casing models.

4.4.7 Nonrectangular Shaped and Non-symmetric Fracture

To test the adaptivity of the CNN to different shaped and non-symmetric fractures, we substitute the rectangular fracture with a composite fracture, including a 300 m × 300 m square at center and a semiellipse of 100-m minor semiaxis on each side as shown in Fig. 4.5; the estimated l will then expect to be around 400 m. The true casing model is used to generate the synthetic scattered fields from the fracture under three orientation angles $\phi = 60^\circ$ (yellow), 90° (green) and 120° (blue), respectively; for the 60° fracture, we further move it by 100 m along its two horizontal extension directions (dashed curves marked either + or -) to make it non-symmetric to its intersection point with casing. White Gaussian noise of different SNRs is then added to test the optimal CNN's (six convolutional layers and two cascaded kernels per layer from Section 4.4.5) performance under different dropout rates.

The estimated ϕ and l are presented in Table 4.3 in the form of mean ± standard deviation of the quintuple results; the testing error for the previous true testing data is also provided as reference. From the table we can read: 1) the CNN works well for nonrectangular and non-symmetric fractures under various noise, demonstrating the feasibility to image arbitrary shaped fracture in the future; 2) testing errors can

Table 4.3: Estimation for nonrectangular and non-symmetric fractures given in ϕ ($^\circ$), l (m) pair

Dropout rate Testing Data	0.1 (323,934)	0.25 (323,934)	0.5 (323,934)
SNR 40 dB	Errors: 7.6, 12.6 60°: 58.9±1.4, 400.7±14.0 90°: 91.4±1.3, 406.7±7.9 120°: 123.0±1.1, 392.9±11.0 60°+: 62.6±1.5, 395.1±10.0 60°-: 53.5±3.7, 389.5±10.6 Noise: 89.4±21.4, 254.6±172.4	Errors: 7.9, 11.7 60°: 58.0±1.9, 395.5±12.5 90°: 92.4±1.6, 399.6±18.2 120°: 121.1±1.8, 398.0±22.8 60°+: 60.7±1.1, 415.8±15.8 60°-: 56.1±3.8, 359.9±18.4 Noise: 83.6±26.0, 270.4±114.3	Errors: 8.4, 18.9 60°: 56.0±1.8, 404.0±19.8 90°: 91.9±1.0, 413.2±15.3 120°: 123.7±2.9, 383.4±26.8 60°+: 58.5±1.3, 409.7±12.6 60°-: 52.5±4.8, 370.0±18.3 Noise: 95.5±22.4, 365.2±81.3
SNR 20 dB	Errors: 9.4, 49.9 60°: 61.1±3.4, 382.5±43.8 90°: 92.8±4.6, 377.3±58.6 120°: 124.2±3.4, 380.7±34.1 60°+: 63.4±1.4, 373.6±54.2 60°-: 55.6±6.9, 343.8±52.3 Noise: 99.1±22.2, 235.1±132.7	Errors: 8.0, 39.5 60°: 56.9±1.3, 349.0±38.4 90°: 91.6±2.1, 379.5±42.7 120°: 122.4±2.1, 357.3±40.4 60°+: 60.2±1.5, 377.8±45.3 60°-: 54.9±3.0, 333.5±59.8 Noise: 79.0±31.1, 319.9±68.0	Errors: 8.6, 44.1 60°: 57.0±2.2, 372.4±26.8 90°: 92.0±1.2, 361.0±36.0 120°: 123.3±2.6, 352.7±32.0 60°+: 59.5±2.2, 372.8±27.9 60°-: 54.3±3.9, 337.2±26.9 Noise: 109.2±26.5, 284.4±56.0
SNR 0 dB	Errors: 21.7, 145.6 60°: 73.1±25.1, 364.5±60.8 90°: 98.4±18.4, 387.5±58.1 120°: 126.3±6.5, 362.0±63.5 60°+: 78.8±23.8, 377.2±81.1 60°-: 69.8±31.0, 347.4±82.6 Noise: 94.5±28.6, 327.3±118.3	Errors: 17.8, 113.2 60°: 60.4±9.7, 316.0±45.8 90°: 93.0±9.7, 284.5±62.2 120°: 124.6±5.9, 310.5±56.0 60°+: 61.0±6.2, 302.9±21.0 60°-: 60.6±6.2, 295.5±44.4 Noise: 116.7±26.5, 344.4±83.3	Errors: 19.5, 113.0 60°: 55.4±5.3, 315.5±36.7 90°: 77.9±12.7, 263.1±35.6 120°: 115.8±13.4, 304.6±50.0 60°+: 57.4±5.8, 323.4±34.1 60°-: 56.8±5.4, 306.1±41.6 Noise: 94.0±31.6, 288.7±89.3

usually serve as a proxy for the performance of the CNN to unseen data, and larger testing errors indicate larger difference between the estimated result and the truth as well as larger standard deviation; 3) the standard deviation is a good proxy for the SNR, and larger standard deviation usually indicates worse SNR; 4) with the deterioration of SNR, the dropout rate needs to synchronously increase to accommodate the noise and achieve the best performance.

Further, we test the cases when no fracture present (as an extreme case of small fractures), by zeroing out the signal but keeping the corresponding noise unchanged at each SNR row in Table 4.3. The corresponding CNN's results are marked as "noise" and show much larger standard deviation than other cases (usually more than four times for ϕ and twice for l), when ignoring the unreliable results from CNN with 0.1 dropout rate for 0 dB noise. This observation suggests us to use the consistency of predictions to judge the detectability of the signal and the confidence of the estimation.

4.5 Improving the Reconstruction Accuracy through Deep Transfer Learning

In the previous sections, we have successfully trained CNNs by mixing true data with approximated data. Although it achieves very good reconstruction accuracy, there is still room to improve, by recognizing that we only need the CNN to reconstruct fractures from true data, rather than from both true and approximated data. To train a CNN specific to true data, we are going to introduce the concept of deep transfer learning to the full wave inversion field for the first time, and the reconstruction results show a notable accuracy improvement.

4.5.1 Deep Transfer Learning

Nowadays, the NNs become deeper and deeper to tackle more and more complicated problems, and therefore require a huge amount of data to train, which consequently increases the computational complexity and the training data preparation. The most straightforward approach to alleviate the long training time issue is by leveraging hardware acceleration, including graphics processing unit (GPU) and tensor processing unit (TPU); however, it also increases the hardware cost. To deal with the performance issues of NNs while keep a sustainable hardware cost, one alternative approach is to capitalize on transfer learning (TL) [Zhuang et al. (2020)].

TL is a technique in ML that focuses on storing knowledge acquired while solving one problem and applying to a different but related problem. The basic idea of the TL is to overcome the isolated learning paradigm and recycle the acquired knowledge. For example, knowledge gained while learning to recognize dogs could apply when trying to recognize cats. From the practical standpoint, recycling or transferring information from previously explored tasks to new tasks has the potential to significantly improve the learning efficiency of new models and reduce the hardware cost.

The deep transfer learning is a combination of DL and TL, where the knowledge of model previously trained by a large-scale data can be applied to a new model. This is very important and useful because the DL model usually has numerous layers and requires a large amount of data to train, which is sometimes impossible or expensive to gather or generate. When the number of training data is relatively small, the model will overfit after several epochs if starting from scratch. On the contrary, if we have a pre-trained model by a large dataset and share some similarity with the current task, then the main body of the model can be reused and only a small portion of the model needs to be tuned by the small dataset, with a lower demand for the computational power.

The most popular applications of the deep transfer learning are natural language processing (NLP) and computer vision (CV). For NLP, [Swietojanski et al. (2012)] dealt with unsupervised cross-lingual knowledge transfer based on a deep NN, and [Huang et al. (2013)] proposed a shared-hidden-layer multi-lingual deep NN to tackle cross-language knowledge TL tasks. For CV, [Kandaswamy et al. (2014)] trained a NN to classify Latin digits and then reused it to classify lowercase letters without starting from scratch. [Chang et al. (2017)] proposed a novel multiscale convolutional sparse coding method to automatically learn filter banks at different scales and provide an unsupervised solution for learning transferable base knowledge and fine-tuning it toward target tasks. [Long et al. (2018)] developed a novel framework for deep adaptation networks that extends deep convolutional NNs to domain adaptation problems and yields state-of-the-art results on standard visual domain-adaptation benchmarks.

4.5.2 Deep Transfer Learning in Full Wave Inversion

Before applying the deep transfer learning in full wave inversion problems, we need to build one source domain and one target domain. In the source domain, specifically, we build a simplified forward model to generate a large amount of approximated data, which can be used to train a NN (left column of Fig. 4.6) of good generalization toward the distorted data space and good reconstruction accuracy toward the model parameters. In the target domain, an ideal NN (right column of Fig. 4.6) needs to be trained by a large amount of high-fidelity data to guarantee good generality toward the realistic data space. By comparing these two NNs in Fig. 4.6, we can find that they share some similarities: 1) two groups of input data have the same dimension and similar patterns, because only fine-structures in the model have been simplified, while transmitter-receiver configurations are kept the same and basic field patterns

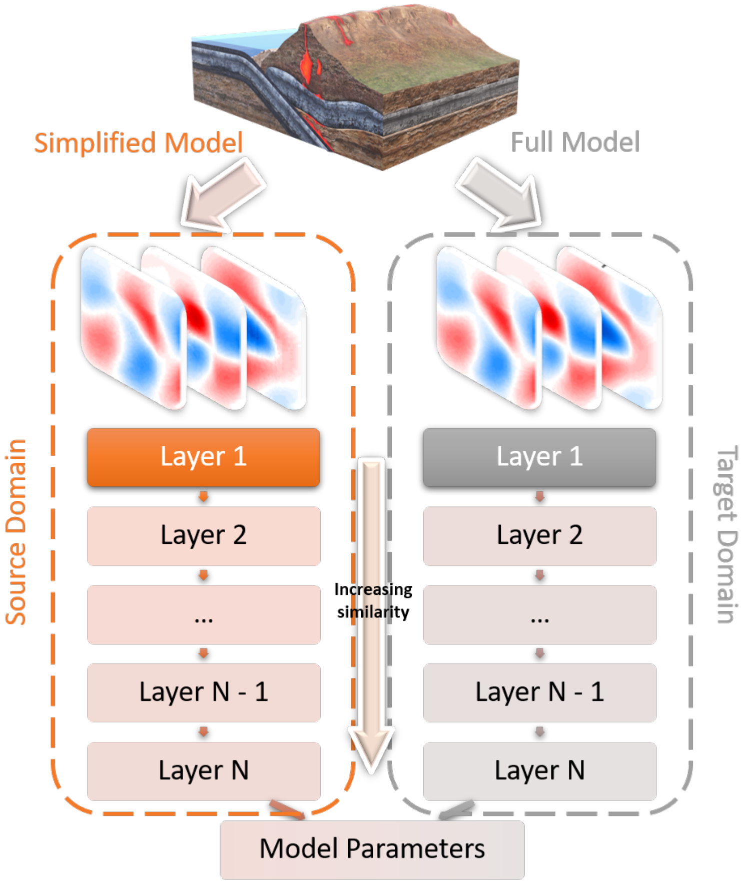


FIGURE 4.6: Diagram of NNs in the source domain and the target domain for FWI

can be captured; 2) two groups of reconstructed model parameters should be exactly the same, because they share the same inversion target; 3) as a result, two groups of layers are expected to converge (become more and more similar) from top down. Therefore, instead of training the NN in realistic data space from scratch, we target to leverage the concept of deep transfer learning: transfer the NN in distorted data space to realistic data space and then fine-tune the top layers, where the layers' weights differ most significantly.

The increasing similarity between NNs from top down is a distinct property of the proposed deep transfer learning in FWI, while the deep transfer learning in CV has the opposite trend [Yosinski et al. (2014)]. In CV tasks, the top layer features that deep NNs tend to learn resemble either Gabor filters or color blobs, which is not only true for different datasets, but also toward different training objectives, like image classification [Krizhevsky et al. (2012); Simonyan and Zisserman (2014); He et al. (2016)], unsupervised learning [Lee et al. (2009)], and unsupervised sparse feature learning [Le et al. (2011)]. Because these standard features on the top layer are independent of the exact loss function and image dataset, top layer features are considered general. On the other hand, the features computed by the last layer must depend greatly on the chosen dataset and task, and are therefore considered specific. The inference is that the middle layers transition from general to specific and the similarity between NNs in source and target domains is decreasing from top down.

4.5.3 Grid Search in Source Domain

To achieve the balance between overfitting and underfitting given the training data, again, we need to tune several hyperparameters of the CNN prototype, including the number of convolutional layers, the number of cascaded convolutional kernels inside convolutional layers, the dropout rate, and the widths of FCLs. From previous study [Zhang et al. (2020)] we can find that, the number of convolutional layers and the number of cascaded convolutional kernels are among the most significant and the most uncertain hyperparameters. On the contrary, the dropout rate (0.25) and the widths of the FCLs (128/16) can have relatively stable optimal settings. By fixing these two hyperparameters, we perform a grid search for the number of convolutional layers over four, five and six (corresponding to a last feature map of $4 \times 4 \times 96$, $2 \times 2 \times 192$ and $1 \times 1 \times 384$), and for the number of cascaded convolutional kernels from one to three.

To test the CNN's performance under different hyperparameter combination, we randomly select 5% of the 1,428 samples as the testing data and divide the rest 1,355 samples into five groups. By taking each group as validation data successively while other four as training data, we can train a CNN quintuple with five independent sets of weights. When evaluating the CNN's performance, the CNN quintuple is applied to the testing data separately to produce five estimations, and the mean value among them will be the final result, while their standard deviation can serve as a proxy for the consistency

To train the CNN quintuple, small batch size (16) is employed to secure good generalization ability, because large-batch methods tend to converge to sharp minimizers of the training function [Keskar et al. (2016)]. The Adam optimizer (learning rate = 0.001), which is based on adaptive estimation of first-order and second-order moments, is used due to its computational efficiency and fast convergence, and mean absolute error (MAE) is employed as the loss function. The training is run on one Tesla K80 GPU from the Google Colab, which normally takes 500 to 1200 epochs to converge. The runtime per epoch largely depends on the network's parameter number, both of which have been shown in Table 4.4 with the format (tunable parameter number, runtime per epoch). Table 4.4 also shows the averaged testing errors with the format as ϕ ($^\circ$), l (m) pair under white Gaussian noise with different signal-to-noise ratios (SNR): no noise, 40 dB, 20 dB and 0 dB. Different channels have added noise of different variances according to their root mean square values to secure the same SNR. Note that in Table 4.4: 1) when noise is large (SNR = 20 dB or 0 dB), the averaged testing errors diverge from the smallest testing errors; therefore, the smallest testing errors are also included as shown in parenthesis; 2) errors are marked as "W" (short for wrong) when they are larger than 30° or 160 m; 3) for CNNs of the same number of kernels per layer, the smallest testing error under 0 dB SNR is marked bold. From Table 4.4 we can read that,

1. CNNs with 2 or 3 cascaded kernels per layer generally have much smaller errors

than those with 1 kernel per layer.

2. CNNs of four convolutional layers show significantly larger errors than the other two candidates, especially when the noise is large.

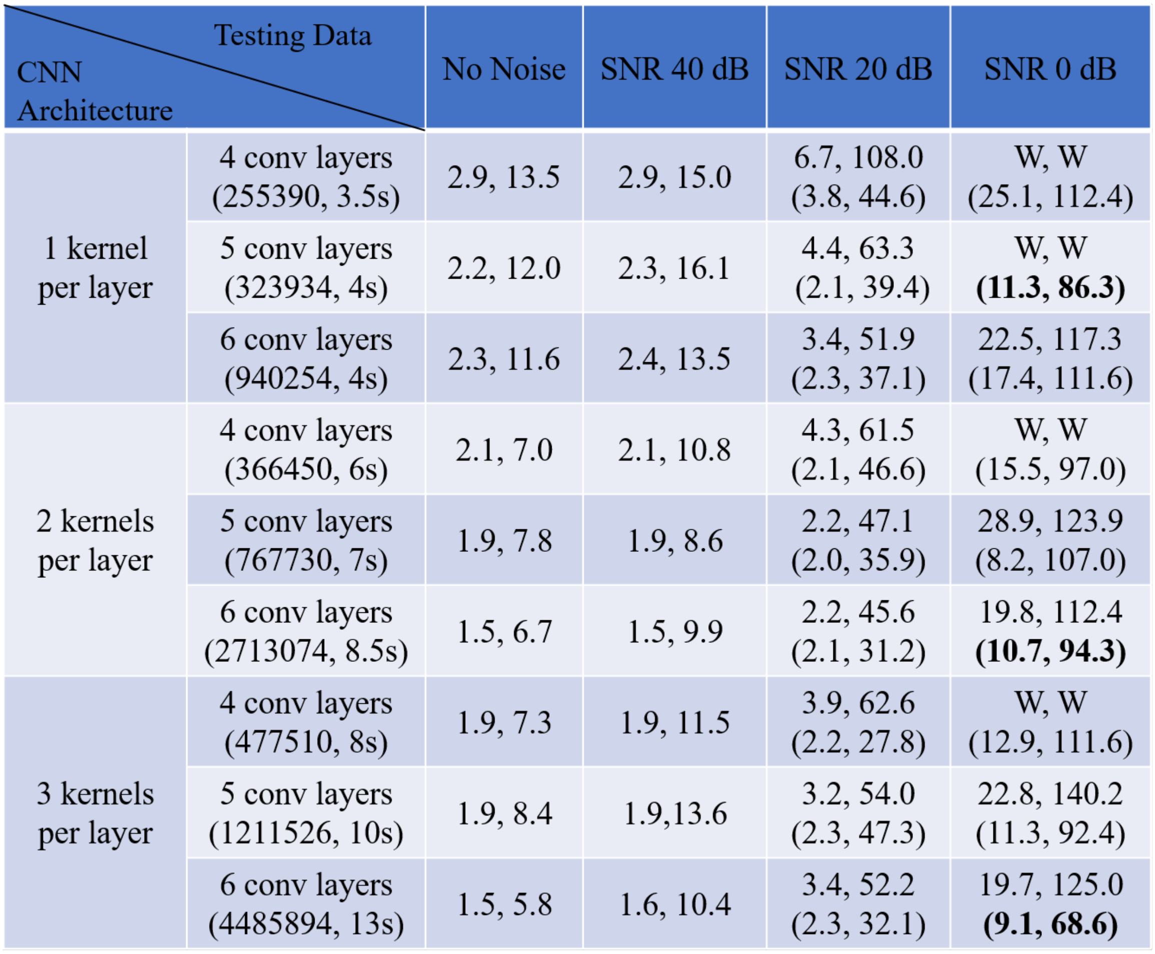
3. The added white Gaussian noise creates some local minimums for the CNN's weights, where the Adam optimizer may end up with. Therefore, the discrepancy among the CNN quintuple may be enlarged along with the increase of noise intensity and the best single CNN may have much smaller error than the averaged error of quintuple.

The averaged error of quintuple is a good indicator for the performance of different hyperparameters. But for the deep transfer learning, we only need to transfer one optimal set of weights to the target domain. Therefore, we select three single CNN candidates, each of which has the optimal CNN configuration and weights for a certain kernel number as marked bold in Table 4.4.

4.5.4 Transfer Scheme in Target Domain

Recall that in our proposed deep transfer learning method, the similarity between CNNs in source and target domains is increasing from top down; therefore, after transferring the well-trained CNN from the source domain to the target domain, we expect to freeze the bottom layers and tune the top layers. One naive approach is to add layers on top of the transferred CNN. However, a careful analysis can show that whether to add FCLs or convolutional layers is not feasible. First, adding a single FCL will introduce tens of millions weights, and the CNN will overfit to the small amount of true data when training. Second, from Fig. 4.7 we can observe that, the approximated scattered field patterns have lower resolution compared to the true scattered field patterns, especially for the tangential components. Since the transferred CNN is trained by the approximated scattered field, it will still expect a

Table 4.4: Testing errors of ϕ (°), l (m) pair in source domain



Testing Data / CNN Architecture		No Noise	SNR 40 dB	SNR 20 dB	SNR 0 dB
1 kernel per layer	4 conv layers (255390, 3.5s)	2.9, 13.5	2.9, 15.0	6.7, 108.0 (3.8, 44.6)	W, W (25.1, 112.4)
	5 conv layers (323934, 4s)	2.2, 12.0	2.3, 16.1	4.4, 63.3 (2.1, 39.4)	W, W **(11.3, 86.3)**
	6 conv layers (940254, 4s)	2.3, 11.6	2.4, 13.5	3.4, 51.9 (2.3, 37.1)	22.5, 117.3 (17.4, 111.6)
2 kernels per layer	4 conv layers (366450, 6s)	2.1, 7.0	2.1, 10.8	4.3, 61.5 (2.1, 46.6)	W, W (15.5, 97.0)
	5 conv layers (767730, 7s)	1.9, 7.8	1.9, 8.6	2.2, 47.1 (2.0, 35.9)	28.9, 123.9 (8.2, 107.0)
	6 conv layers (2713074, 8.5s)	1.5, 6.7	1.5, 9.9	2.2, 45.6 (2.1, 31.2)	19.8, 112.4 **(10.7, 94.3)**
3 kernels per layer	4 conv layers (477510, 8s)	1.9, 7.3	1.9, 11.5	3.9, 62.6 (2.2, 27.8)	W, W (12.9, 111.6)
	5 conv layers (1211526, 10s)	1.9, 8.4	1.9,13.6	3.2, 54.0 (2.3, 47.3)	22.8, 140.2 (11.3, 92.4)
	6 conv layers (4485894, 13s)	1.5, 5.8	1.6, 10.4	3.4, 52.2 (2.3, 32.1)	19.7, 125.0 **(9.1, 68.6)**

relatively rough input; however, if adding convolutional layers, it will be trained by sharp images and the output will be much smoother than what the transferred CNN expects, no matter how their weights are tuned.

Instead of adding more layers on top of the transferred CNN, we keep the CNN architecture invariant, but fine-tune the parameters of several top layers in the target domain. Since the true scattered field pattern is similar to the approximated scattered field pattern, we can use the well-trained weights from the source domain as the initial value for the top layers and gradually adapt them to the true data. To keep

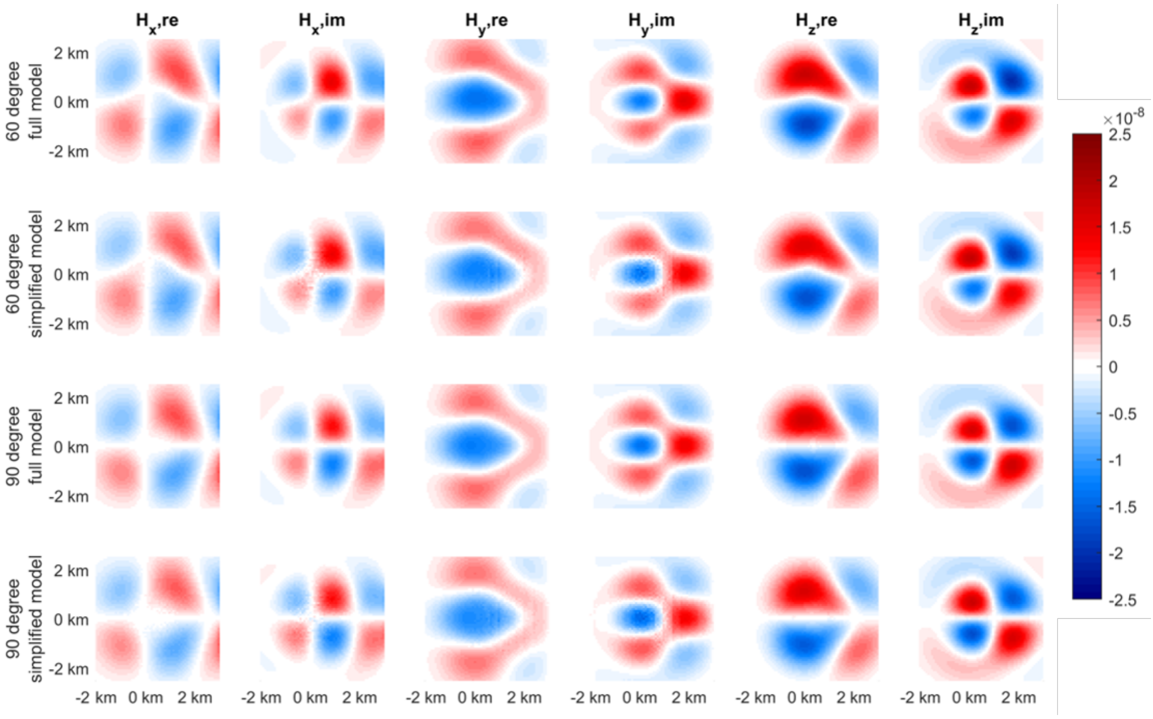
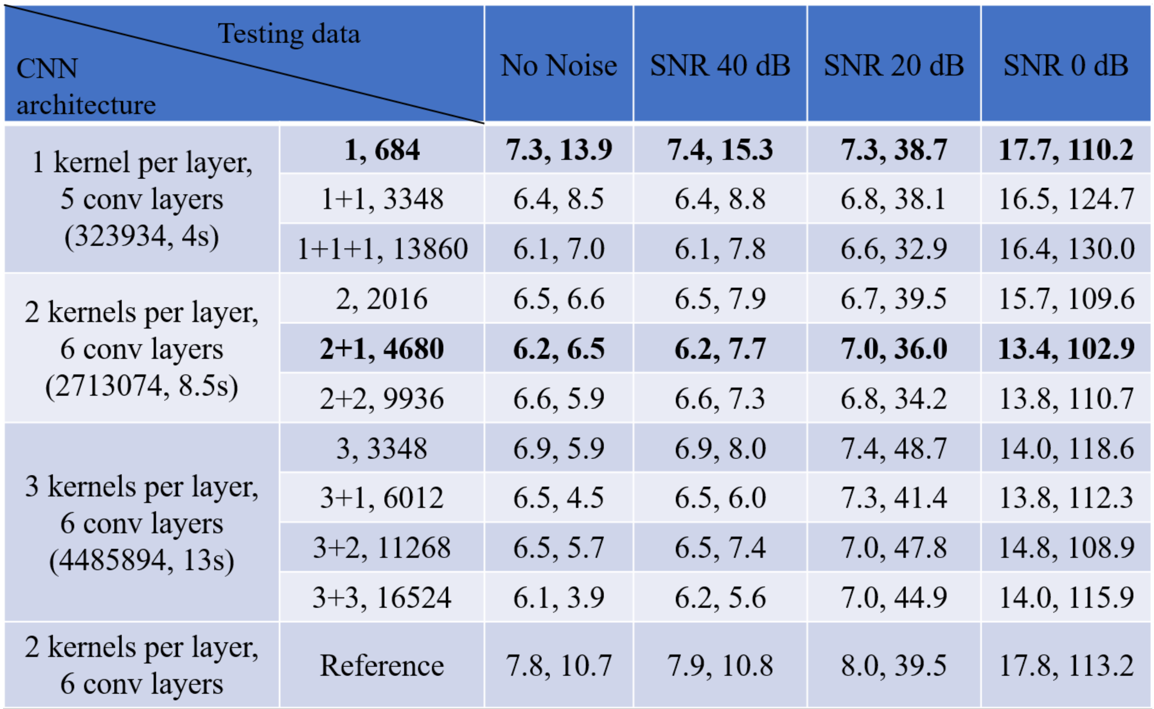


FIGURE 4.7: First (second) row: scattered magnetic fields (A/m) from a hydraulic fracture of $l = 500$ m, $\phi = 60°, \sigma = 500$ S/m using the full (simplified) model. Third (fourth) row: scattered magnetic fields (A/m) from a hydraulic fracture of $l = 500$ m, $\phi = 90°, \sigma = 500$ S/m using the full (simplified) model.

a short data generation time, only 54 true samples, fewer than 5% of approximated samples, are employed to fine-tune the model, which correspond to fractures of $\phi =$ 20, 45, 75, 105, 135, 160 degree, l =100, 300, 500 m and σ = 100, 300, 500 S/m. The true samples are also divided into five groups to train a CNN quintuple in the target domain, with the same training settings as those in the source domain. To provide a fair comparison with the CNN quintuple trained in [Zhang et al. (2020)], the same testing data is employed, including 75 samples from fractures of ϕ = 30, 60, 90, 120, 150 degree, l =100, 300, 500 m and σ = 100, 200, 300, 400, 500 S/m, where σ is up sampled to test the CNN's generalization ability.

To determine how to divide the transferred CNN into top and bottom parts, we employ a grid search for the number of kernels in top part. The averaged testing

Table 4.5: Testing errors of ϕ (°), l (m) pair in target domain



Testing data CNN architecture		No Noise	SNR 40 dB	SNR 20 dB	SNR 0 dB
1 kernel per layer, 5 conv layers (323934, 4s)	**1, 684**	**7.3, 13.9**	**7.4, 15.3**	**7.3, 38.7**	**17.7, 110.2**
	1+1, 3348	6.4, 8.5	6.4, 8.8	6.8, 38.1	16.5, 124.7
	1+1+1, 13860	6.1, 7.0	6.1, 7.8	6.6, 32.9	16.4, 130.0
2 kernels per layer, 6 conv layers (2713074, 8.5s)	2, 2016	6.5, 6.6	6.5, 7.9	6.7, 39.5	15.7, 109.6
	2+1, 4680	**6.2, 6.5**	**6.2, 7.7**	**7.0, 36.0**	**13.4, 102.9**
	2+2, 9936	6.6, 5.9	6.6, 7.3	6.8, 34.2	13.8, 110.7
3 kernels per layer, 6 conv layers (4485894, 13s)	3, 3348	6.9, 5.9	6.9, 8.0	7.4, 48.7	14.0, 118.6
	3+1, 6012	6.5, 4.5	6.5, 6.0	7.3, 41.4	13.8, 112.3
	3+2, 11268	6.5, 5.7	6.5, 7.4	7.0, 47.8	14.8, 108.9
	3+3, 16524	6.1, 3.9	6.2, 5.6	7.0, 44.9	14.0, 115.9
2 kernels per layer, 6 conv layers	Reference	7.8, 10.7	7.9, 10.8	8.0, 39.5	17.8, 113.2

errors are given in Table 4.5, where the data format of the second column is (number of tunable kernels, number of tunable parameters), and the addend from left to right represents the number of tunable convolutional kernels at each convolutional layer from top down. Notice that the number of tunable parameters in the target domain is much smaller than that in the source domain, and therefore the training time in the target domain is negligible. From the Table 4.5 we can read that,

1. Compared to the testing errors in the source domain, the errors in target domain generally grow a little.

2. The CNN candidate of 1 kernel per layer has significantly larger errors than other two candidates, especially under 0 dB noise.

3. The CNN candidate of 2 kernels per layer and 2+1 tunable kernels (both two

kernels in the first convolutional layer and the first kernel in the second convolutional layer) has the smallest errors overall, and has notably smaller errors than the optimal CNN trained by data augmentation as shown in the bottom row of Table 4.5. Note that the comparison between them is fair because they share the CNN architecture.

4. The CNN candidate of 1 kernel per layer and 1 tunable kernel (the only one in the first convolutional layer) has similar errors as the optimal CNN trained by data augmentation, but with a much smaller architecture, and therefore can save more than half of the training time.

4.5.5 Analysis of the Deep Transfer Learning's Performance

To better understand the optimal CNN's (2 kernels per layer, 6 convolutional layers and 2+1 tunable kernels) performance toward different fractures under different noise, we plot averaged reconstructed fracture parameters along with the ground truth in a 2D space spanned by l and ϕ in Fig. 4.8. The ground truth of 75 true samples has been degenerated into 15 blue square grid points in each subplot, due to the overlapping of points with the same l and ϕ but different σ. The reconstructed results have been presented by markers with different shapes, sizes and colors. Specifically, reconstructed results with different ϕ are distinguished by shapes ($\lhd, \circ, \triangle, \star, \rhd$ for 30°, 60°, 90°, 120°, and 150°), results with different l are distinguished by sizes (larger markers for larger l), and results with different σ are distinguished by colors as shown by the colorbar. From Fig. 4.8 we can observe that,

1. For the cases with no noise or low noise (SNR = 40 dB), the CNN quintuple can reconstruct the fractures for different ϕ, l, and σ with a good agreement with the ground truth.

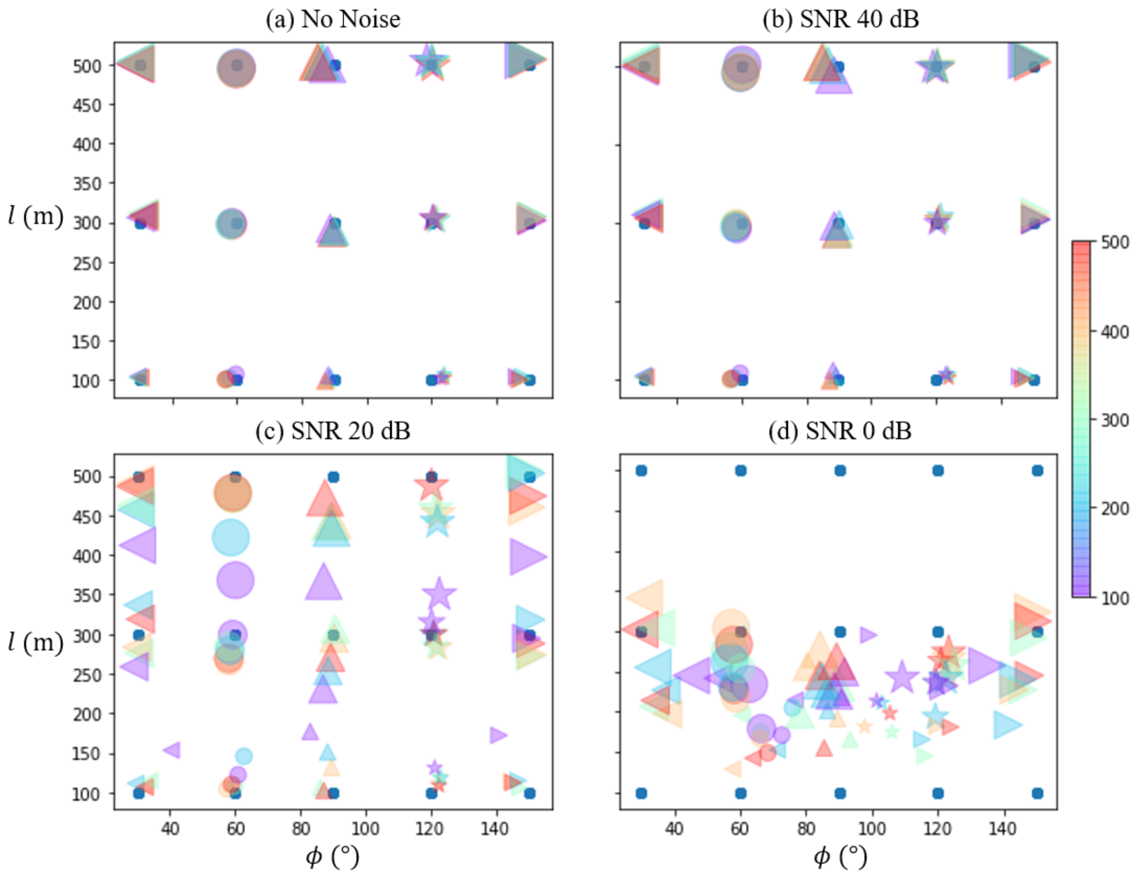


FIGURE 4.8: Scatter plots of 75 fractures with (a) no noise, (b) 40 dB noise, (c) 20 dB noise, and (d) 0 dB noise. 15 blue square grid points are the ground truth for the fractures in the ϕ-l plane. The shape, size and color of markers represent their truth values of ϕ, l, and σ, while their coordinates indicate the reconstructed ϕ and l.

2. For the cases with moderate noise (SNR = 20 dB), the reconstructed results start to diverge from the ground truth, but mainly along the vertical direction, indicating the estimation of ϕ is more robust than that of l. Besides, the estimation of fractures with lower σ suffers from larger errors, due to corresponding weaker scattering fields.

3. For the cases with high noise (SNR = 0 dB), the length of long fractures (l = 500 m) have been underestimated significantly, but the reconstructed ϕ

is relatively accurate. On the contrary, the inversion of short fractures (l = 100 m) suffer from extreme uncertainty for both ϕ and l. In a nutshell, the uniformly distributed scatter plot has shrunk significantly, indicating the CNN quintuple's discriminability has reduced due to high noise.

4. Except for the generally larger errors for lower σ, the estimation for fractures of σ = 200 S/m and 400 S/m is no worse than the estimation for others, demonstrating good generalization toward unseen conductivity.

4.6 Summary

In summary, this chapter presents a systematic approach to image hydraulic fractures under the energized steel casing. A borehole-to-surface system is employed to excite the fracture at close range and collect the scattered fields on the ground. An approximated casing model is proposed to accelerate the synthetic data generation, followed by the error and sensitivity analysis, which demonstrates that magnetic fields are more accurate in approximated casing model and more discriminative in the fracture imaging than the electric fields for the presented configuration. A model-based inversion method is then proposed with a careful dataset construction, a CNN architecture specification and two training/testing designs. Grid search is applied to investigate CNNs' performance on different hyperparameters, where the optimal hyperparameters can generalize the fracture conductivity, accommodate model approximation errors, and therefore reduce by 90% (50%) of the KNN's length (azimuthal angle) error towards unseen conductivity. The well-trained CNN also shows robust estimation under white Gaussian noise, especially for the orientation. Moreover, we include an example for nonrectangular and non-symmetric fracture imaging, which shows the probability of a two-step inversion for more complicated fractures: first obtain a precise estimate of fractures' orientation and dimensions, and then reconstruct its detailed geometry. Finally, we introduce the deep transfer learning to the full wave inversion problem for the first time and further improve the reconstruction accuracy. The proposed deep transfer learning-based inversion method can be generalized to many other simulation-based ML/DL applications by following a four-step approach:

1. Build a simplified physical model to accelerate/enable the synthetic data generation and produce a large amount of approximated data.
2. Design a NN and train it with the approximated data in the source domain.
3. Build a full physical model to generate a small amount of true data.
4. Transfer the NN to the target domain and fine-tune the parameters of top layers through the true data.

To extend the CNN-based inversion method to other inversion applications, the interested application should include grid receivers or line receivers to enable the convolution operation; further, the application should expect a real-time process to embody the CNN's advantage and provide a fixed experimental setup and environment in advance to allow data generation and model training.

5

Devine Field Data Inversion through Deep Learning

Hydraulic fracturing has enabled oil and gas extraction from low-permeability reservoirs. Although this technology has advanced very rapidly in recent years, reliable diagnoses and appraisal of various fracture properties (dimensions, orientation, conductivity, and proppant distribution) suffer from extreme measurement uncertainties. Here we explore the possibility to use the EM geophysical sensing data, which is measured through a cross-borehole system with galvanic stimulation at Devine Test Site. By leveraging the NN [Hornik et al. (1989); Araya-Polo et al. (2017)], we propose a near real-time inversion method to reconstruct the proppant distribution in a hydraulic fracture at the Devine Test Site in this chapter. Towards that end, we first interpolate the proppant distribution with a B-spline function, whose control points are randomized to generate different shapes. After assigning various conductivities to proppants, we employ a FEM-based EM fast solver to model the fracture scattering problem and collect a large number of synthetic data to train, validate and test our NN. The input dimension of the NN is tailored to fit the number of "valid" field data points, where the field data point is recognized as "valid" if it locates within the range of corresponding synthetic data points recorded by the same TR pair for dif-

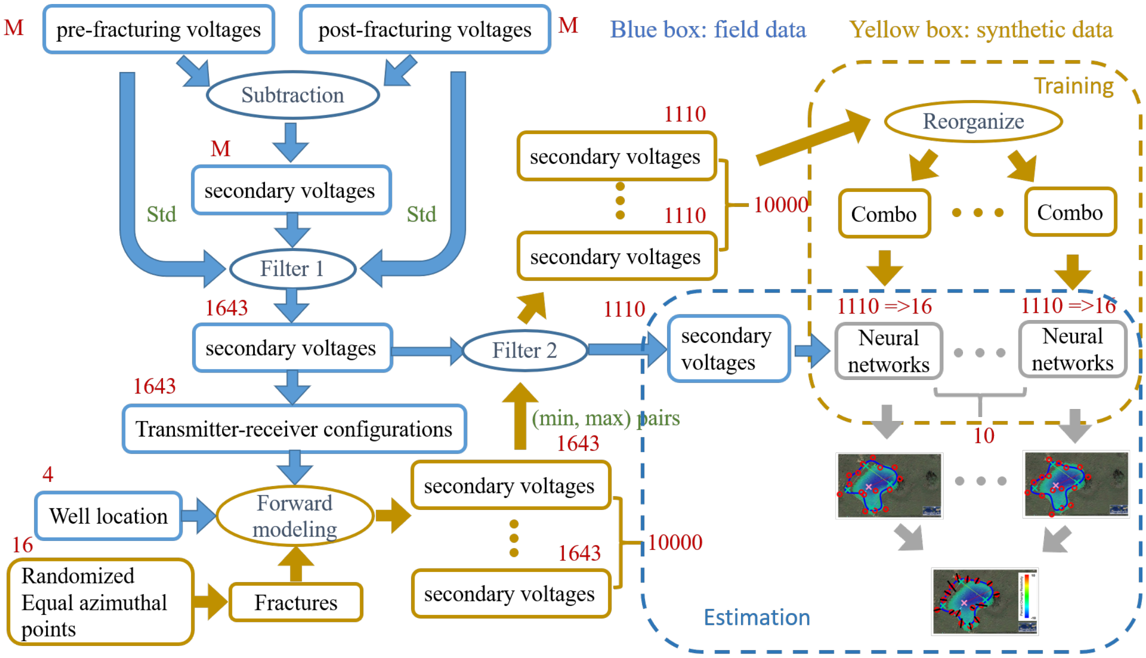


FIGURE 5.1: Workflow of the whole inversion process has been shown here, where the rectangular boxes represent data or models, elliptical boxes represent operators, and red letters/numbers represent the dimension of corresponding data/objects. Filter 1 is to delete the secondary voltages smaller than the std of either the pre-fracturing or post-fracturing voltages and filter 2 is to delete the secondary voltages outside of the corresponding (min, max) pairs.

ferent proppant distributions/conductivities. Tenfold cross validation is then used to train ten NNs independently and obtain ten reconstructions, whose mean values can provide a robust point estimate and variance can provide a measure of confidence.

5.1 Experimental Setup and Methods

The whole workflow of the inversion has been plotted in Fig. 5.1, including preprocess/filtering, forward modeling, training/validation, and estimation, which is demonstrated in detail in the following.

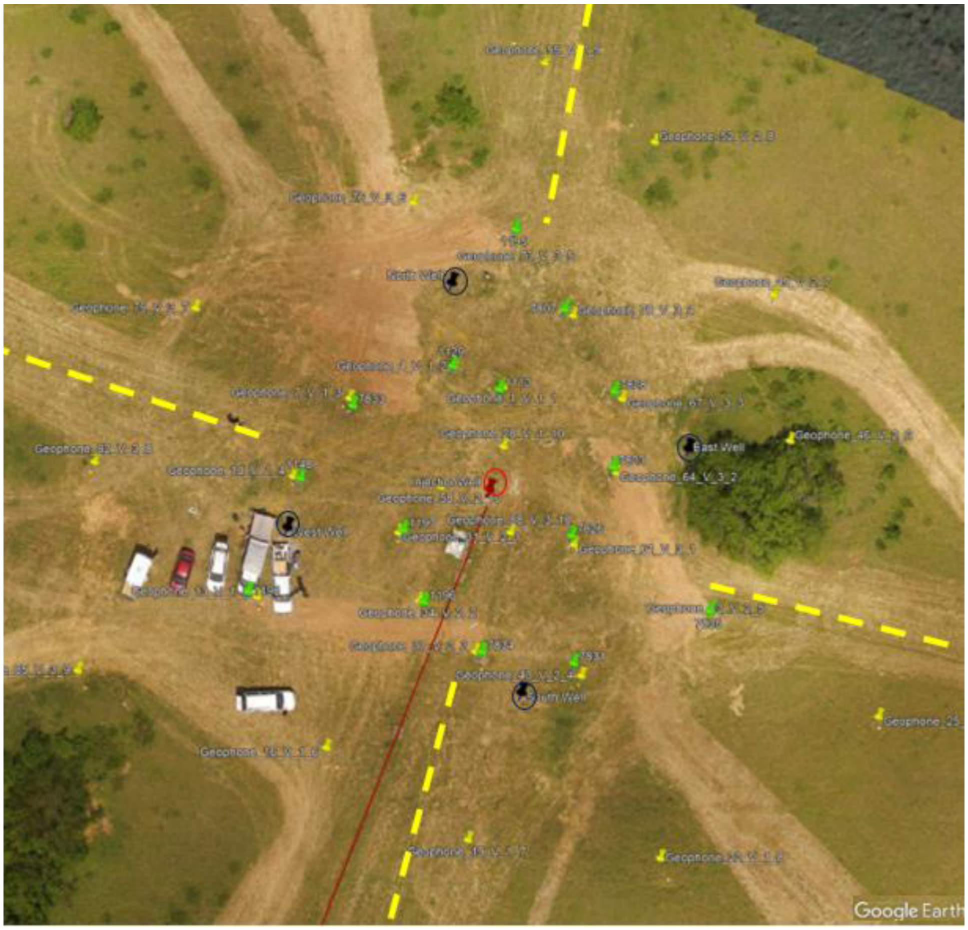
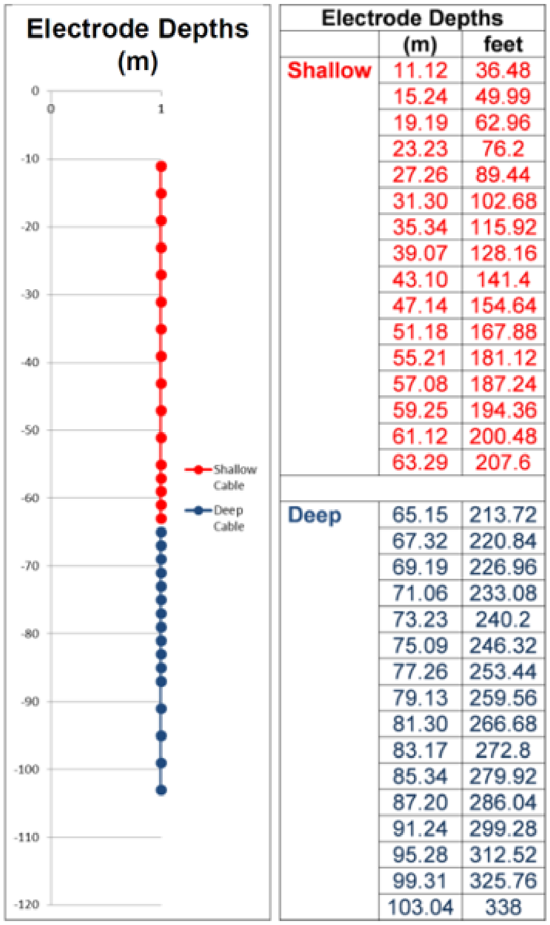


FIGURE 5.2: Aerial view of AEC's inverted five-spot well pattern at Devine Test Site. Injection well shown with red circle; OBs shown with black circle; surface ERT arrays shown with yellow dashed lines; surface geophones and tilt-vault locations shown by small yellow and green pins, respectively; service line connecting fracture truck to injection well shown with red line in southwest direction [Ahmadian et al. (2018)].

5.1.1 Experimental Setup and Preprocess

In April 2017, the AEC successfully completed data collection for a proof-of-concept demonstration of remote mapping of hydraulically fractured networks using EM proppant additives and a variety of EM tools and configurations at the Devine Test Site, located approximately 50 miles southwest of San Antonio, Texas, and managed by the Bureau of Economic Geology (Bureau) at The University of Texas at Austin.



Electrode Depths		
	(m)	feet
Shallow	11.12	36.48
	15.24	49.99
	19.19	62.96
	23.23	76.2
	27.26	89.44
	31.30	102.68
	35.34	115.92
	39.07	128.16
	43.10	141.4
	47.14	154.64
	51.18	167.88
	55.21	181.12
	57.08	187.24
	59.25	194.36
	61.12	200.48
	63.29	207.6
Deep	65.15	213.72
	67.32	220.84
	69.19	226.96
	71.06	233.08
	73.23	240.2
	75.09	246.32
	77.26	253.44
	79.13	259.56
	81.30	266.68
	83.17	272.8
	85.34	279.92
	87.20	286.04
	91.24	299.28
	95.28	312.52
	99.31	325.76
	103.04	338

FIGURE 5.3: Cable layout inside the OBs [Ahmadian et al. (2018)]

The detailed experimental tools and configurations can be found in [Ahmadian et al. (2018)]. Here we only focus on the tool and configuration that collect the time-domain-induced polarization (TDIP) data we used in this chapter. An inverted five-spot vertical well pattern as shown in Fig. 5.2 was completed to a depth of 104 m, where the center injection well was completed by a 10-cm steel casing for hydraulic stimulation with an electrically conductive contrast agent at a depth of about 53.3 m (174.9 ft), and four observation wells (OBs) were constructed in each of the four corners, but cased with a 10-cm polyvinyl chloride (PVC) casing. The OBs were equipped with arrays of electrodes for electromagnetic resistivity tomography and electromagnetic impedance tomography (ERT/EIT). Two borehole ERT cables were installed in each OB, allowing them to be treated as a single, continuous 32-electrode line as shown in Fig. 5.3. The electrode spacing at 4 m allowed the ERT cable to collect data from a depth of 10 m (33 ft) at the surface to 103 m (338 ft) at the bottom of the well. The 2-m spacing ranged from a depth of 55.3 m (181.6 ft) to 87.3 m (286.5 ft) enabled better data resolution near the targeted injection points (located at 60 m (197 ft) and 70 m (230 ft)). TDIP method-based data acquisition used the wires to connect the measurement system with the electrodes; constant currents in these wires were turned on or off to produce square-wave signals and then their 2-Hz component were extracted.

To obtain the secondary voltage from the hydraulic fracture, pre-fracturing and post-fracturing voltages were collected before and after the hydraulic fracturing, respectively. For each of them, multiple TR combinations were measured reciprocally, because reciprocals usually provide a better estimate of data accuracy than simple repetitions of the same TR combination. Pre-fracturing/post-fracturing voltages with relative reciprocal error larger than 2% were deleted; calculated secondary voltages (difference between the post-fracturing and pre-fracturing voltages) with amplitude smaller than either their pre-fracturing or post-fracturing voltages' standard deviation were also deleted. Finally, 1,643 data points from 1,643 TR pairs were left for the next step.

5.1.2 Fracture Modeling

Previous voxel-based inversion and tiltmeter survey show that the hydraulic fracture at the Devine Test Site is almost horizontal at a depth of 54 m and inside a 30-m-radius circle with the injection well as the center [Ahmadian et al. (2018)]; therefore, we restrict the proppant distribution within a 2D horizontal circle in this chapter as shown in Fig. 5.4. Note that to restrict the proppants to a specific planar or

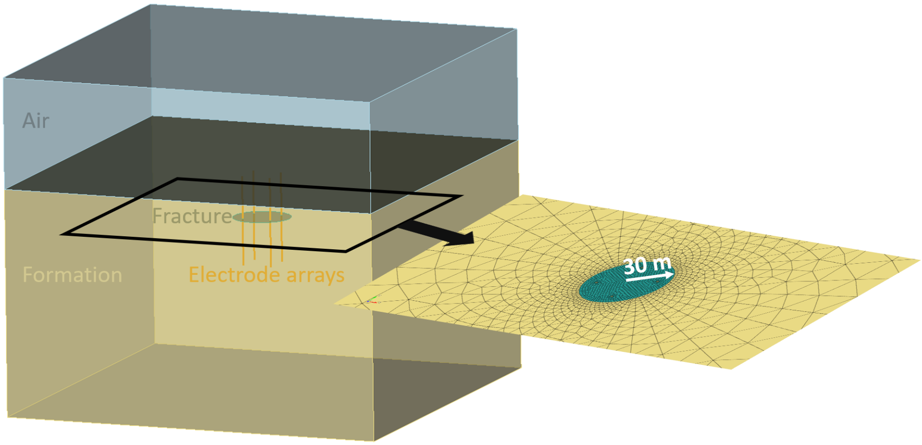


Figure 5.4: The configuration for the Devine Test Site; the mesh for the inner part of fracture plane is given at the right, where the 30-m-radius green circle is the largest possible proppant distribution considered here.

curved surface is reasonable prior information, because in many cases we target to monitor the proppant distribution when injecting proppants to pre-existing hydraulic fractures with known position and extent. Further, since proppants are always distributed in a connected region instead of a cluster of isolated regions when being pumped or just finished, we characterize the distribution with a parametric closed curve and try to invert the curve's parameters instead of independent pixels, which tremendously reduces the number of unknowns and accelerates the training process. Specifically, the proppant distribution is modeled by a 4th-order B-spline function of 16 control points, each of which is constrained to one of the equiangular radial lines as shown in Fig. 5.5. By adjusting the distances between the control points and the center through 16 normalized lengths (between 0.2 and 1), we can fit the closed curve to arbitrary shapes. Consequently, we have 16 numbers to invert. The lower bound 0.2 is set empirically by realizing that there is no direction where the proppants do not flow at all.

To generate the training data, we first need to randomize proppant distributions.

Since in reality the 16 control points are not independent, we apply a Markov chain process to generate the randomized shapes as following,

1. Randomly select one of 16 control points and sample its normalized length from $N(\mu = 0.6, \sigma = 0.4)$;

2. Select the clockwisely adjacent control point and sample its normalized length from $N(\mu = s_p, \sigma = 0.4)$, where s_p is the previous control point's sampled normalized length; repeat this process until the last control point.

3. Sample the last control point's normalized length from $N(\mu = (s_p + s_1)/2, \sigma = 0.4)$, where s_1 is the first control point's normalized length.

4. For all samples from Gaussian distribution $N(\mu, \sigma)$, if the chosen normalized length is larger than 1 or smaller than 0.2, simply resample it until it lies within the acceptable range.

Besides, the conductivity profile of proppants is also empirically simplified as a three-piece constant function of the distance to the injection well. As shown in Fig. 5.5, three similar contours are respectively 100%, 75% and 50% of the randomized shape with fixed conductivity relationship $\sigma_{inner,red} = 2\sigma_{middle,yellow} = 5\sigma_{outer,green}$. In the following, when we mention conductivity σ it will refer to $\sigma_{inner,red}$.

After generating 2,000 groups of control points for 2000 shapes, we modify our FEM-based fast EM solver to accelerate the synthetic secondary voltage generation as following, which typically takes four minutes per case on a 12-core Intel Haswell standard compute node [Towns et al. (2014)].

1. Generate one single tetrahedral mesh grid for the Devine Test Site, where the fracture is modeled as a 30-m-radius horizontal circle densely; the region

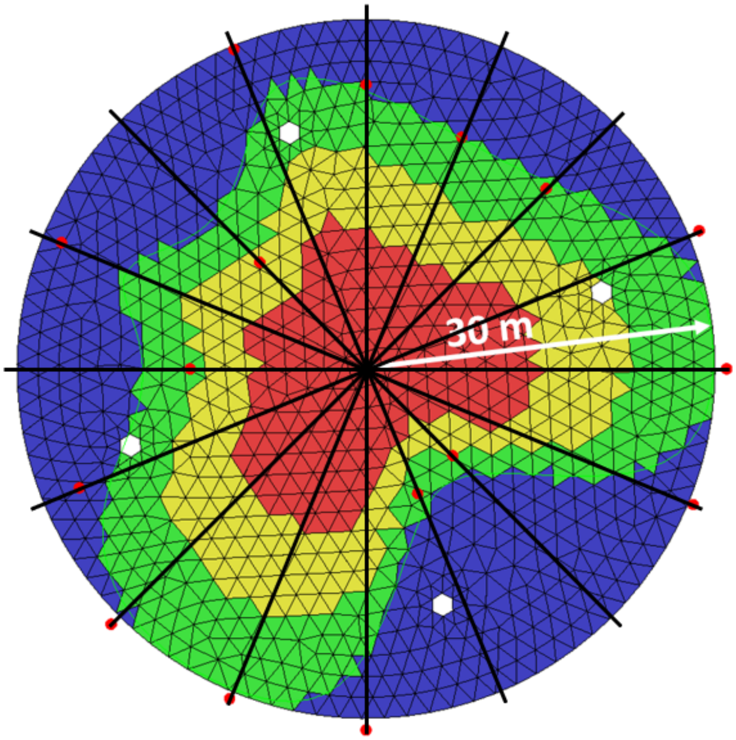


FIGURE 5.5: One randomly generated proppant distribution with prefixed mesh and piecewise constant conductivity of $\sigma_{inner,red} = 2\sigma_{middle,yellow} = 5\sigma_{outer,green}$. 16 control points (red) are restricted to corresponding radial lines (black).

of electrode arrays is also meshed densely; all the surrounding regions are meshed coarsely, which includes a top region for air and a bottom region for homogeneous formation of 0.02 S/m conductivity (estimated from resistivity log [Ahmadian et al. (2018)]). The total degrees of freedom (DoF) is more than 1 million.

2. Preprocess the mesh grid and assemble the mass matrix, stiffness matrix as well as the scattering boundary condition (SBC) matrix into the global matrix.

3. Traverse the Cartesian product of 2,000 control point groups and five proppant conductivities σ (1, 2, 5, 10, 20 S/m) to perform the following three steps:

   (a) Employ one group of control points to interpolate a B-spline closed curve, which is then used to divide the 30-m-radius circle into two parts: triangular elements outside (inside) the closed curve are assigned with background (proppant) conductivity;

   (b) Model the whole circle with the impedance transmission boundary condition (ITBC) [Sun et al. (2017); Zhang et al. (2020)] and integrate the ITBC matrix into the global matrix;

   (c) Solve post-injection voltage drop over all 1,643 TR pairs together efficiently by the parallel sparse direct solver Intel MKL PARDISO;

4. Set the whole 30-m-radius circle's conductivity as background and compute the voltage drops over 1,643 TR pairs as pre-injection voltages; subtract the pre-injection voltages from post-injection voltages of every proppant distribution to get the secondary voltages.

Note that five conductivities have been employed to augment the synthetic data, which is to balance the NN's generality and the cost of synthetic data generation. If we fix the conductivity, the amount of synthetic data will reduce by 80%, but the trained NN may overfit to this specific conductivity and cannot perform well for the real case. If we further increase the number of investigated conductivities, the conductivity can then be included as another target parameter together with the 16 normalized lengths. However, the amount of synthetic data and therefore the data generation time will increase significantly, while the inverted conductivity is much less useful than reconstructed shape.

5.1.3 Neural Network Configuration

A multi-layer fully connected NN is employed here, which is expected to explore the intrinsic relationship between the measured voltage drops and the control points' normalized lengths. Each fully connected layer includes one linear mapping layer with ReLU activation, one batch normalization layer and probably one dropout layer. The linear mapping layer with ReLU activation is the nonlinear foundation for the NN; the batch normalization is to adjust the mean (standard deviation) of output close to 0 (1) to facilitate the learning process; the dropout layer disables part of the neurals randomly to increase the noise robustness. Multiple fully connected layers of the same dimension form a fully connected unit. The input dimension of the NN is determined by calculating the 1,643 ranges of corresponding TRs from 10,000 groups of synthetic data. If the field data point sits between its corresponding range, it will be fed to the NN; otherwise, it will be abandoned. This process can help to filter out the field data point which has a large discrepancy from the synthetic data points due to either noise pollution or complicated real formation. The input dimension is then shrunk to 1,110. The limitation of this filtering process is that it can only be employed after the acquisition and preprocess of the field data. Therefore, the training of NN needs to be included in the total inversion time and is very critical to achieve a near real-time inversion.

To transform 1,110 inputs to 16 outputs, we design a five-unit fully connected NN with dimensions as powers of two from 2^9 to 2^5. The input layer and fully connected layers in the top unit are equipped with dropout layers with dropout rates 0.3 and 0.1, respectively, because larger dimension can support and benefit more from higher dropout rate.

To provide a measure of confidence, after randomly selecting 5% of the 10,000 groups of synthetic data as testing data, we use a tenfold cross validation by splitting the remaining 9,500 groups of synthetic data into ten sets and taking each set as validation data successively and other nine as training data. Consequently, we will acquire ten independent sets of weights for the NN, which will then predict ten independent proppant distributions for the same field data. Finally, we can calculate one standard deviation per control point from ten reconstructed distributions to estimate the confidence level.

5.2 Results and Discussion

To guarantee the near real-time inversion, we target to train ten NNs in 20 minutes. Since training processes for the NNs are independent to each other, they can be done concurrently and each training can have 20 minutes to accomplish. By balancing the training time and accuracy on a 24-core Intel Haswell standard compute node, we determine the number of fully connected layers in one unit to be two and employ the Adam optimizer with learning rate of 0.01 and the mean absolute error (MAE) loss function. The batch size is set as 512 and the number of epochs is 2000. Larger batch size cannot obtain expected validation error and smaller batch size will increase the training time per epoch. The mean values of training/validation/testing errors for the well trained NNs are around 0.045/0.055/0.050. The corresponding reconstructed proppant distributions from the field data by the ten NNs are shown in Fig. 5.6. By calculating the mean and standard deviation for each of the control points from the ten groups, a final reconstructed shape has been provided in Fig. 5.7, where the red circles are the mean values and the black lines range from mean-std to mean+std. Shorter lines indicate smaller standard deviation as well as higher confidence level. From Fig. 5.7 we can conclude that, reconstructed shapes are consistent (shorter black lines) and also agree well with the reference at the top right part of the fracture, where the traditional inversion method [Ahmadian et al. (2018)] indicates a higher conductivity (dark blue). For the bottom part, the reconstructed shapes have some inconsistencies (longer black lines) and are smaller than the reference, because the lower conductivity of the potential proppants (the large light blue region in the reference) is hard to capture. For the left top part, the reconstructed shapes are also not consistent and larger than the reference.

To evaluate the similarity between the NN result and the reference, we use the

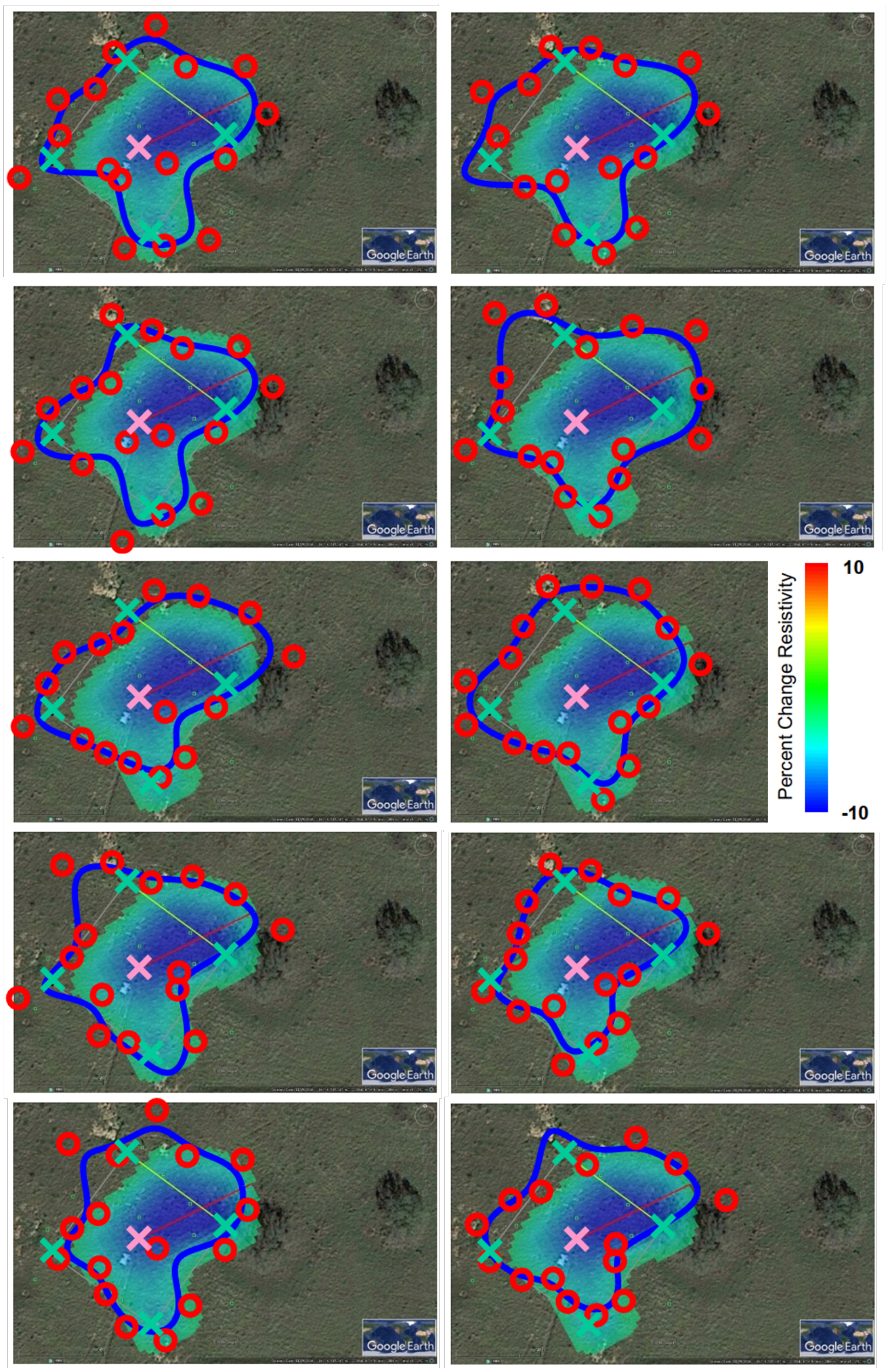


FIGURE 5.6: Ten reconstructed proppant distributions from the NNs. The red circles are control points inverted by the NNs and the blue curves are interpolated using the B-spline function. The pink cross is the injection well and the green crosses are OBs. The colorful conductivity contour is the reference from voxel-based inversion [Ahmadian et al. (2018)].

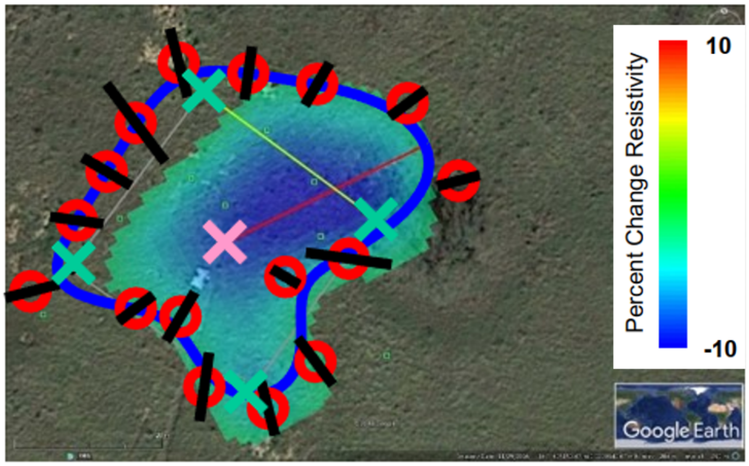


FIGURE 5.7: The averaged reconstructed proppant distribution from the NNs. The red circles are the mean values of ten groups of control points inverted by the NNs and the blue curve is interpolated using the B-spline function. The black lines range from mean-std to mean+std. The pink cross is the injection well and the green crosses are OBs. The colorful conductivity contour is the reference from voxel-based inversion [Ahmadian et al. (2018)].

Intersection over Union (IoU, also known as the Jaccard similarity coefficient) as the metric, which is widely used in computer vision for comparing the similarity of two sample sets. IoU is defined as the intersection area divided by the union area of the comparison pair A and B:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{5.1}$$

In our case IoU ≈ 0.74, which is a large number and indicates a good agreement between the NN and the reference.

However, if we do not have a reference, it will be hard for us to evaluate the reconstruction's accuracy as well as the NN's performance, because the conventional data misfit is hard to calculate and also not representative in this case: 1. the proppant conductivity is not the reconstruction target and therefore unknown; 2. the NN fundamentally tries to discover the intrinsic relationship between the pattern of secondary voltages and the distribution of proppants, and therefore the reconstructed shape does not necessarily guarantee a lower data misfit; 3. the NN suppresses the noise impact through the dropout layer, but it is not clear how to reduce the noise impact when calculating the data misfit; therefore, the calculated data misfit will include both the error and the noise. One alternative solution is to use the confidence level to shed some light on the reconstruction accuracy, as discussed above.

5.3 Summary

The fully connected NN has been successfully applied to map the electrically active proppants in near real time. By filtering out erroneous field data using the synthetic data range and employing the tenfold cross validation, we can guarantee an accurate and robust reconstruction. The final reconstructed shape is determined by the mean of the ten results and the confidence intervals of different parts can be estimated by corresponding control points' standard deviations. The technology proposed in this chapter paves the way for near real-time remote monitoring of dynamic changes of proppants in hydraulically fractured networks.

6

Conclusions and Future Work

6.1 Conclusions

Based on the DGFD method, one hierarchical sudoku mesh and one flipped TF/SF DGFD method have been proposed to accurately and efficiently model the vertical well logging and the LWD problem. The hierarchical sudoku mesh inherits the gradient property from the progressive tetrahedron mesh and the high-order structured mesh from the SEM, and then combines them by the DGFD-based DDM. Because the hierarchical sudoku mesh's specific spatial discretization conforms to the diffusion pattern and the geometrical characteristics of the logging problem with as few unknowns as possible, it provides both fast convergence and short running time in each iteration step. Numerical cases demonstrate that with the help of hierarchical sudoku mesh, the DGFD-based DDM, and theoretically all DDM schemes, can have a better performance when modeling vertical well logging problem over the traditional FEM. The flipped TF/SF DGFD method can help to avoid both the excessive meshes around the source and the elementwise integration of the scattering objects for the moving tool. By honoring the invariance of the system matrix, the flipped

TF/SF DGFD method can handle multiple RHSs all at once with PARDISO. Numerical examples show that the flipped TF/SF DGFD method can save up to 70% unknowns, which increases the computing capability for the LWD application under the same computing resource greatly. For long wellbore trajectories of thousands meters, it is reasonable to break the whole trajectory into independent small problems; the flipped TF/SF DGFD method can then help to enlarge the computational domain for each small problem, and correspondingly reduce the number of independent small problems.

Based on the DL, two different types of NNs have been designed to reconstruct the hydraulic fracture under two scenarios. The first scenario is to reconstruct the hydraulic fractures under the energized steel casing, and a borehole-to-surface system has been employed by exciting the fracture at close range and collecting the scattered fields on the ground. An approximated casing model has been proposed to accelerate the synthetic data generation by enlarging the casing radius and reducing its conductivity. To improve the accuracy and robustness of the reconstruction as well as capture the most important characters of the fracture, direction and lateral extent of the fracture have been used to parameterize the fracture and serve as the target parameters of the inversion. Two training/testing designs have been explored and the optimized hyperparamters after grid search can generalize the fracture conductivity, accommodate model approximation errors, and therefore reduce by 90% (50%) of the KNN's length (azimuthal angle) error towards unseen conductivity. The well-trained CNN also shows robust estimation under white Gaussian noise, especially for the orientation. Moreover, after imaging a nonrectangular and non-symmetric fracture, a two-step inversion for more complicated fractures becomes promising: first obtain a precise estimate of fractures' orientation and dimensions, and then reconstruct its detailed geometry. To further improve the reconstruction accuracy, we introduce the deep transfer learning to the full wave inversion problem for the first time and demonstrate its potential generalization toward simulation-based ML/DL applications. To extend the CNN-based inversion method to other inversion applications, the interested application should include grid receivers or line receivers to enable the convolution operation; further, the application should expect a real-time process to embody the CNN's advantage and provide a fixed experimental setup and environment in advance to allow data generation and model training.

The second scenario is to reconstruct the shape of proppant distribution inside a fracture from a cross-borehole system with galvanic stimulation at Devine Test Site. To reduce the number of unknowns to reconstruct, the proppant distribution is characterized as a B-spline function interpolated 2D closed curve with 16 control points. Randomized proppant distributions modeled by the ITBC have been used to generate a large amount of synthetic data through a FEM-based EM fast solver. Bootstrapping technique has been employed to train multiple NNs given different subsets of the whole training set, which can then reconstruct multiple proppant distribution candidates. The mean value of them has been proved to provide an accurate and robust inversion result in agreement with the reference and the variance can estimate the uncertainty of different orientations of the inversion result.

6.2 Future Work

The following work of this dissertation will focus on the DL application in both forward modeling and inverse problem. Due to the fast mapping capability, the DL finds great opportunities on both forward modeling and inverse problem.

In this dissertation, some progress has been made towards the DL-based fracture imaging. For the steel casing stimulated fracture detection, the next plan is to reconstruct multiple fractures' direction and lateral extent, as well as to further reconstruct the fracture's shape. For the Devine field data inversion, the next plan is to reconstruct the time-variant proppant distributions by using the field data measured at different moments.

Although the DL-based forward modeling has not been discussed in this dissertation, it is within the plan of future study. The advantage of the DL-based forward modeling is also its fast solution phase compared to the conventional forward solvers, which is critical to inversion and optimization that involves multiple forward iterations. The shortage of the DL-based forward modeling is its poor versatility; in other words, the application will need to be specific to guarantee a good solution accuracy. I propose to start from the subsurface detection case with simple parametric geometries to achieve a good accuracy as well as fast speed and then apply the DL-based forward solver to inverse problems.

Bibliography

Aarre, V., Astratti, D., Al Dayyni, T. N. A., Mahmoud, S. L., Clark, A. B., Stellas, M. J., Stringer, J. W., Toelle, B., Vejbæk, O. V., and White, G. (2012), "Seismic detection of subtle faults and fractures," *Oilfield review*, 24, 28–43.

Ahmadian, M., LaBrecque, D., Liu, Q. H., Slack, W., Brigham, R., Fang, Y., Banks, K., Hu, Y., Wang, D., Zhang, R., et al. (2018), "Demonstration of Proof of Concept of Electromagnetic Geophysical Methods for High Resolution Illumination of Induced Fracture Networks," in *SPE Hydraulic Fracturing Technology Conference and Exhibition*, Society of Petroleum Engineers.

Araya-Polo, M., Dahlke, T., Frogner, C., Zhang, C., Poggio, T., and Hohl, D. (2017), "Automated fault detection without seismic processing," *The Leading Edge*, 36, 208–214.

Augustin, A., Kennedy, W., Morrison, H. F., and Lee, K. (1989), "A theoretical study of surface-to-borehole electromagnetic logging in cased holes," *Geophysics*, 54, 90–99.

Bruce, A., Wong, P., Zhang, Y., Salisch, H., Fung, C., and Gedeon, T. (2000), "A state-of-the-art review of neural networks for permeability prediction," *The APPEA Journal*, 40, 341–354.

Calvetti, D. and Somersalo, E. (2018), "Inverse problems: From regularization to Bayesian inference," *Wiley Interdisciplinary Reviews: Computational Statistics*, 10, e1427.

Chang, H., Han, J., Zhong, C., Snijders, A. M., and Mao, J.-H. (2017), "Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications," *IEEE transactions on pattern analysis and machine intelligence*, 40, 1182–1194.

Chaumont-Frelet, T., Pardo, D., and Rodríguez-Rozas, Á. (2018), "Finite element simulations of logging-while-drilling and extra-deep azimuthal resistivity measurements using non-fitting grids," *Computational Geosciences*, 22, 1161–1174.

Chave, A. D. and Jones, A. G. (2012), *The magnetotelluric method: Theory and practice*, Cambridge University Press.

Chen, J. (2017), "An efficient layered finite element method with domain decomposition for simulations of resistivity well logging," *Journal of Petroleum Science and Engineering*, 150, 217–223.

Chen, S., Chew, W., Santos, V., Sainath, K., and Teixeira, F. (2016), "Electromagnetic subsurface remote sensing," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–28.

Cipolla, C., Wright, C., et al. (2000), "State-of-the-art in hydraulic fracture diagnostics," in *SPE Asia Pacific Oil and Gas Conference and Exhibition*, Society of Petroleum Engineers.

Cockburn, B., Karniadakis, G. E., and Shu, C.-W. (2000), "The development of discontinuous Galerkin methods," in *Discontinuous Galerkin Methods*, pp. 3–50, Springer.

Commer, M., Hoversten, G. M., and Um, E. S. (2015), "Transient-electromagnetic finite-difference time-domain earth modeling over steel infrastructure," *Geophysics*, 80, E147–E162.

Constable, S. (2010), "Ten years of marine CSEM for hydrocarbon exploration," *Geophysics*, 75, 75A67–75A81.

Cuevas, N. H. (2014), "Analytical solutions of EM fields due to a dipolar source inside an infinite casing," *Geophysics*, 79, E231–E241.

da Silva, N. V., Morgan, J. V., MacGregor, L., and Warner, M. (2012), "A finite element multifrontal method for 3D CSEM modeling in the frequency domain," *Geophysics*, 77, E101–E115.

Dai, J., Fang, Y., Zhou, J., and Liu, Q. H. (2018), "Analysis of electromagnetic induction for hydraulic fracture diagnostics in open and cased boreholes," *IEEE Transactions on Geoscience and Remote Sensing*, 56, 264–271.

Daniels, D. J. (1996), "Surface-penetrating radar," *Electronics & Communication Engineering Journal*, 8, 165–182.

Danielsen, J. E., Auken, E., Jørgensen, F., Søndergaard, V., and Sørensen, K. I. (2003), "The application of the transient electromagnetic method in hydrogeophysical surveys," *Journal of applied geophysics*, 53, 181–198.

Davis, T. A., Rajamanickam, S., and Sid-Lakhdar, W. M. (2016), "A survey of direct methods for sparse linear systems," *Acta Numerica*, 25, 383–566.

De Castro, L. N. (2006), *Fundamentals of natural computing: basic concepts, algorithms, and applications*, Chapman and Hall/CRC.

Doll, H. et al. (1949), “Introduction to induction logging and application to logging of wells drilled with oil base mud,” *Journal of Petroleum Technology*, 1, 148–162.

Fang, Y., Dai, J., Yu, Z., Zhou, J., and Liu, Q. H. (2017), “Through-casing hydraulic fracture evaluation by induction logging I: An efficient EM solver for fracture detection,” *IEEE Transactions on Geoscience and Remote Sensing*, 55, 1179–1188.

Fang, Y., Hu, Y., Zhan, Q., and Liu, Q. H. (2018), “Electromagnetic Forward and Inverse Algorithms for 3-D Through-Casing Induction Mapping of Arbitrary Fractures,” *IEEE Geoscience and Remote Sensing Letters*, 15, 996–1000.

Fang, Y., Hu, Y., Zhan, Q., Wang, D., Zhang, R., and Liu, Q.-H. (2019), “A Fast Numerical Method for the Galvanic Measurement in Hydraulic Fracture Detection,” *IEEE Transactions on Antennas and Propagation*, pp. 1 – 12.

Fletcher, R. (2013), *Practical methods of optimization*, John Wiley & Sons.

Gonzalez, J. A. (2013), “A discontinuous Galerkin finite element method for the time-domain solution of Maxwell equations,” Ph.D. thesis, Universidad de Granada.

Goodfellow, I., Bengio, Y., and Courville, A. (2016), *Deep learning*, MIT press.

Grayver, A. V. and Bürg, M. (2014), “Robust and scalable 3-D geo-electromagnetic modelling approach using the finite element method,” *Geophysical Journal International*, 198, 110–125.

He, K., Zhang, X., Ren, S., and Sun, J. (2016), “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hong, D., Xiao, J., Zhang, G., and Yang, S. (2014), “Characteristics of the sum of cross-components of triaxial induction logging tool in layered anisotropic formation,” *IEEE Transactions on Geoscience and Remote Sensing*, 52, 3107–3115.

Hornik, K., Stinchcombe, M., White, H., et al. (1989), “Multilayer feedforward networks are universal approximators.” *Neural networks*, 2, 359–366.

Hoversten, G. M., Commer, M., Haber, E., and Schwarzbach, C. (2015), “Hydro-frac monitoring using ground time-domain electromagnetics,” *Geophysical Prospecting*, 63, 1508–1526.

Hu, Y., Wang, G. L., Liang, L., and Abubakar, A. (2017), “Estimation of reservoir parameters from inversion of triaxial induction data constrained by mud-filtrate invasion modeling,” *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, 2, 228–236.

Hu, Y., Fang, Y., Wang, D., Zhong, Y., and Liu, Q. H. (2018), "Electromagnetic waves in multilayered generalized anisotropic media," *IEEE Transactions on Geoscience and Remote Sensing*, 56, 5758–5766.

Huang, J.-T., Li, J., Yu, D., Deng, L., and Gong, Y. (2013), "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7304–7308, IEEE.

Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. (2017), "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, 26, 4509–4522.

Kandaswamy, C., Silva, L. M., Alexandre, L. A., Santos, J. M., and de Sá, J. M. (2014), "Improving deep neural network performance by reusing features trained with transductive transference," in *International Conference on Artificial Neural Networks*, pp. 265–272, Springer.

Karlsson, A. (2009), "Approximate Boundary Conditions for Thin Structure," 57, 144–148.

Keller, G. V. and Frischknecht, F. C. (1966), "Electrical methods in geophysical prospecting," .

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016), "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*.

King, G. E. et al. (2010), "Thirty years of gas shale fracturing: What have we learned?" in *SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers.

Komatitsch, D. and Tromp, J. (1999), "Introduction to the spectral element method for three-dimensional seismic wave propagation," *Geophysical journal international*, 139, 806–822.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012), "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105.

Le, Q. V., Karpenko, A., Ngiam, J., and Ng, A. Y. (2011), "ICA with reconstruction cost for efficient overcomplete feature learning," in *Advances in neural information processing systems*, pp. 1017–1025.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009), "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th annual international conference on machine learning*, pp. 609–616.

Lee, J. and Peng, Z. (2017), *Computational Electromagnetics: Domain Decomposition Methods and Practical Applications*, CRC Press.

Lee, J.-H. and Liu, Q. H. (2007), "A 3-D spectral-element time-domain method for electromagnetic simulation," 55, 983–991.

Lee, J.-H., Xiao, T., and Liu, Q. H. (2006), "A 3-D spectral-element method using mixed-order curl conforming vector basis functions for electromagnetic fields," *IEEE transactions on Microwave Theory and Techniques*, 54, 437–444.

Li, Y.-J. and Jin, J.-M. (2007), "A new dual-primal domain decomposition approach for finite element simulation of 3-D large-scale electromagnetic problems," *IEEE Transactions on Antennas and Propagation*, 55, 2803–2810.

Long, M., Cao, Y., Cao, Z., Wang, J., and Jordan, M. I. (2018), "Transferable representation learning with deep adaptation networks," *IEEE transactions on pattern analysis and machine intelligence*, 41, 3071–3085.

Løseth, L. and Ursin, B. (2007), "Electromagnetic fields in planarly layered anisotropic media," *Geophysical Journal International*, 170, 44–80.

Luo, M., Lin, Y., and Liu, Q. H. (2013), "Spectral methods and domain decomposition for nanophotonic applications," *Proceedings of the IEEE*, 101, 473–483.

Ma, J., Nie, Z., and Sun, X. (2014a), "Efficient modeling of large-scale electromagnetic well-logging problems using an improved nonconformal FEM-DDM," *IEEE Transactions on Geoscience and Remote Sensing*, 52, 1825–1833.

Ma, J., Jin, J.-M., and Nie, Z. (2014b), "A nonconformal FEM-DDM With tree–cotree splitting and improved transmission condition for modeling subsurface detection problems," *IEEE Transactions on Geoscience and Remote Sensing*, 52, 355–364.

Martin, L. E. S., Fouda, A. E., Amineh, R. K., Capoglu, I., Donderici, B., Roy, S. S., Hill, F., et al. (2017), "New High-Definition Frequency Tool for Tubing and Multiple Casing Corrosion Detection," in *SPE Abu Dhabi International Petroleum Exhibition & Conference*, Society of Petroleum Engineers.

Maxwell, S. (2014), *Microseismic imaging of hydraulic fracturing: Improved engineering of unconventional shale reservoirs*, Society of Exploration Geophysicists.

McCann, M. T., Jin, K. H., and Unser, M. (2017), "A review of convolutional neural networks for inverse problems in imaging," *arXiv preprint arXiv:1710.04011.*

Mitzner, K. (1968), “Effective boundary conditions for reflection and transmission by an absorbing shell of arbitrary shape,” *IEEE Transactions on Antennas and propagation*, 16, 706–712.

Nédélec, J.-C. (1980), “Mixed finite elements in 3,” *Numerische Mathematik*, 35, 315–341.

Nikravesh, M. (2007), “Computational intelligence for geosciences and oil exploration,” in *Forging New Frontiers: Fuzzy Pioneers I*, pp. 267–332, Springer.

Nocedal, J. and Wright, S. (2006), *Numerical optimization*, Springer Science & Business Media.

Palisch, T., Al-Tailji, W., Bartel, L., Cannan, C., Zhang, J., Czapski, M., Lynch, K., et al. (2018), “Far-field proppant detection using electromagnetic methods: latest field results,” *SPE Production & Operations*, 33, 557–568.

Pardo, D. and Torres-Verdín, C. (2013), “Sensitivity analysis for the appraisal of hydrofractures in horizontal wells with borehole resistivity measurements,” *Geophysics*, 78, D209–D222.

Pardo, D. and Torres-Verdín, C. (2015), “Fast 1D inversion of logging-while-drilling resistivity measurements for improved estimation of formation resistivity in high-angle and horizontal wells,” *Geophysics*, 80, E111–E124.

Pardo, D., Demkowicz, L., Torres-Verdín, C., and Paszynski, M. (2006), “Simulation of resistivity logging-while-drilling (LWD) measurements using a self-adaptive goal-oriented hp-finite element method,” *SIAM Journal on Applied Mathematics*, 66, 2085–2106.

Pardo, D., Demkowicz, L., Torres-Verdin, C., and Paszynski, M. (2007), “A self-adaptive goal-oriented hp-finite element method with electromagnetic applications. Part II: Electrodynamics,” *Computer methods in applied mechanics and engineering*, 196, 3585–3597.

Paszyński, M., Pardo, D., Torres-Verdín, C., Demkowicz, L., and Calo, V. (2010), “A parallel direct solver for the self-adaptive hp Finite Element Method,” *Journal of Parallel and Distributed Computing*, 70, 270–281.

Peng, Z. and Lee, J.-F. (2012), “A Scalable Nonoverlapping and Nonconformal Domain Decomposition Method for Solving Time-Harmonic Maxwell Equations in Rˆ3,” *SIAM Journal on Scientific Computing*, 34, A1266–A1295.

Peterson, A. F. (1988), “Absorbing boundary conditions for the vector wave equation,” *Microwave and Optical Technology Letters*, 1, 62–64.

Peterson, A. F., Ray, S. L., Mittra, R., of Electrical, I., and Engineers, E. (1998), *Computational methods for electromagnetics*, vol. 2, IEEE press New York.

Puzyrev, V. (2019), “Deep learning electromagnetic inversion with convolutional neural networks,” *Geophysical Journal International*, 218, 817–832.

Puzyrev, V., Koldan, J., de la Puente, J., Houzeaux, G., Vázquez, M., and Cela, J. M. (2013), “A parallel finite-element method for three-dimensional controlled-source electromagnetic forward modelling,” *Geophysical Journal International*, 193, 678–693.

Puzyrev, V., Koric, S., and Wilkin, S. (2016), “Evaluation of parallel direct sparse linear solvers in electromagnetic geophysical problems,” *Computers & geosciences*, 89, 79–87.

Puzyrev, V., Vilamajo, E., Queralt, P., Ledo, J., and Marcuello, A. (2017), “Three-dimensional modeling of the casing effect in onshore controlled-source electromagnetic surveys,” *Surveys in Geophysics*, 38, 527–545.

Puzyrev, V., Torres-Verdín, C., and Calo, V. (2018), “Interpretation of deep directional resistivity measurements acquired in high-angle and horizontal wells using 3-D inversion,” *Geophysical Journal International*, 213, 1135–1145.

Ren, Y., Huang, W.-F., Liu, Q. H., Chen, Y. P., and Zhang, H.-S. (2016), “Accurate fracture scattering simulation by thin dielectric sheet-based surface integral equation,” *IEEE Geosci. Remote Sens. Lett.*, 13, 1448–1451.

Ren, Y., Chen, Y., Zhan, Q., Niu, J., and Liu, Q. H. (2017), “A higher order hybrid SIE/FEM/SEM method for the flexible electromagnetic simulation in layered medium,” *IEEE Transactions on Geoscience and Remote Sensing*, 55, 2563–2574.

Rogers, S. J., Fang, J., Karr, C., and Stanley, D. (1992), “Determination of lithology from well logs using a neural network (1),” *AAPG bulletin*, 76, 731–739.

Rutledge, J. T. and Phillips, W. S. (2003), “Hydraulic stimulation of natural fractures as revealed by induced microearthquakes, Carthage Cotton Valley gas field, east TexasHydraulic Stimulation of Natural Fractures,” *Geophysics*, 68, 441–452.

Saad, Y. (2003), *Iterative methods for sparse linear systems*, vol. 82, siam.

Schenk, O. and Gärtner, K. (2004), “Solving unsymmetric sparse systems of linear equations with PARDISO,” *Future Generation Computer Systems*, 20, 475–487.

Seydoux, J., Legendre, E., Mirto, E., Dupuis, C., Denichou, J.-M., Bennett, N., Kutiev, G., Kuchenbecker, M., Morriss, C., Yang, L., et al. (2014), “Full 3D deep directional resistivity measurements optimize well placement and provide reservoir-scale imaging while drilling,” in *SPWLA 55th Annual Logging Symposium*, Society of Petrophysicists and Well-Log Analysts.

Shahriari, M., Pardo, D., Picón, A., Galdrán, A., Del Ser, J., and Torres-Verdín, C. (2018), "A Deep Learning Approach to the Inversion of Borehole Resistivity Measurements," *arXiv preprint arXiv:1810.04522*.

Simonyan, K. and Zisserman, A. (2014), "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*.

Sleijpen, G. L. and Fokkema, D. R. (1993), "BiCGstab (l) for linear equations involving unsymmetric matrices with complex spectrum," *Electronic Transactions on Numerical Analysis*, 1, 2000.

Sun, Q. (2017), "Discontinuous Galerkin Based Multi-Domain Multi-Solver Technique for Efficient Multiscale Electromagnetic Modeling," Ph.D. thesis, Duke University.

Sun, Q., Zhang, R., Zhan, Q., and Liu, Q. H. (2017), "Multiscale hydraulic fracture modeling with discontinuous Galerkin frequency-domain method and impedance transition boundary condition," *IEEE Transactions on Geoscience and Remote Sensing*, 55, 6566–6573.

Sun, Q., Zhan, Q., Zhang, R., and Liu, Q. H. (2018), "Mixed Total Field/Scattered Field-Based Discontinuous Galerkin Frequency-Domain Method for Subsurface Sensing," *IEEE Transactions on Geoscience and Remote Sensing*, 57, 3354–3360.

Sun, X. Y. and Nie, Z.-P. (2008), "Vector finite element analysis of multicomponent induction response in anisotropic formations," *Progress In Electromagnetics Research*, 81, 21–39.

Swietojanski, P., Ghoshal, A., and Renals, S. (2012), "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, pp. 246–251, IEEE.

Tafti, T. A., Sahimi, M., Aminzadeh, F., and Sammis, C. G. (2013), "Use of microseismicity for determining the structure of the fracture network of large-scale porous media," *Physical Review E*, 87, 032152.

Tarantola, A. (1987), "Inverse problem theory: Method for data fitting and model parameter estimation," *Elsevier*, 613.

Tietze, K., Ritter, O., and Veeken, P. (2015), "Controlled-source electromagnetic monitoring of reservoir oil saturation using a novel borehole-to-surface configuration," *Geophysical Prospecting*, 63, 1468–1490.

Tikhonov, A. N. (1963), "Solution of incorrectly formaulated problems and the regularization method," in *Dokl. Akad. Nauk.*, vol. 151, pp. 1035–1038.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., Roskies, R., Scott, J. R., and Wilkins-Diehr, N. (2014), “XSEDE: Accelerating Scientific Discovery,” *Computing in Science & Engineering*, 16, 62–74.

Travassos, X. L., Avila, S. L., and Ida, N. (2018), “Artificial Neural Networks and Machine Learning Techniques applied to Ground Penetrating Radar: A Review,” *Applied Computing and Informatics.*

Um, E. S., Commer, M., Newman, G. A., and Hoversten, G. M. (2015), “Finite element modelling of transient electromagnetic fields near steel-cased wells,” *Geophysical Journal International*, 202, 901–913.

Wang, H., Barber, T. D., Morriss, C., Rosthal, R. A., Chen, K.-C., Smits, J. W., Minerbo, G. N., Frey, M. T., Homan, D. M., Davydycheva, S., and Tumbiolo, G. (2006), “Triaxial induction logging: theory, modeling, inversion and Interpretation,” in *International Oil & Gas Conference and Exhibition in China*, Society of Petroleum Engineers.

Wang, H., Huang, W.-F., Fang, Y., Zhang, R., Wang, D., Zhan, Q., and Liu, Q. H. (2019), “Efficient and Accurate Electromagnetic Modeling of Triaxial Induction Responses from Multiscale Fractures for Well Logging Applications,” *IEEE Journal on Multiscale and Multiphysics Computational Techniques.*

Wang, L., Li, H., Fan, Y., and Wu, Z. (2018), “Sensitivity analysis and inversion processing of azimuthal resistivity LWD measurements,” *Journal of Geophysics and Engineering.*

Ward, S. H., Hohmann, G. W., and Nabighian, M. (1988), “Electromagnetic theory for geophysical applications,” in *Electromagnetic methods in applied geophysics*, vol. 1, pp. 131–311.

Wei, Z. and Chen, X. (2018), “Deep-learning schemes for full-wave nonlinear inverse scattering problems,” *IEEE Transactions on Geoscience and Remote Sensing*, 57, 1849–1860.

Weiss, C. J., Aldridge, D. F., Knox, H. A., Schramm, K. A., and Bartel, L. C. (2016), “The direct-current response of electrically conducting fractures excited by a grounded current source,” *Geophysics*, 81, 201–210.

Wu, Z., Fan, Y., Wang, J., Zhang, R., and Liu, Q. H. (2019), “2.5D Finite Difference Modeling of Logging-While-Drilling Electromagnetic Measurements in Complex Scenarios,” *IEEE Geoscience and Remote Sensing Letters.*

Yang, K., Torres-Verdín, C., and Yılmaz, A. E. (2015), “Detection and quantification of three-dimensional hydraulic fractures with horizontal borehole resistivity measurements," *IEEE Transactions on Geoscience and Remote Sensing*, 53, 4605–4615.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014), "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, pp. 3320–3328.

Yu, Z., Zhou, J., Fang, Y., Hu, Y., and Liu, Q. H. (2017), "Through-casing hydraulic fracture evaluation by induction logging II: The inversion algorithm and experimental validations," *IEEE Transactions on Geoscience and Remote Sensing*, 55, 1189–1198.

Zeng, S., Li, D., Wilton, D. R., and Chen, J. (2018a), "Fast and accurate simulation of electromagnetic telemetry in deviated and horizontal drilling," *Journal of Petroleum Science and Engineering*, 166, 242–248.

Zeng, S., Chen, F., Li, D., Chen, J., and Chen, J. (2018b), "A novel 2.5 D finite difference scheme for simulations of resistivity logging in anisotropic media," *Journal of Applied Geophysics*, 150, 144–152.

Zhang, R., Sun, Q., Wu, Z., Fang, Y., Hu, Y., Huang, W.-F., and Liu, Q. H. (2019a), "Fast Induction Logging Modeling With Hierarchical Sudoku Meshes Based on DGFD," *IEEE Geoscience and Remote Sensing Letters*, 16, 1683–1687.

Zhang, R., Wu, Z., Sun, Q., Zhuang, M., Cai, Q.-M., Wang, D., and Liu, Q. H. (2019b), "Memory-efficient 3-D LWD solver with the flipped total field/scattered field-based DGFD method," *IEEE Geoscience and Remote Sensing Letters.*

Zhang, R., Sun, Q., Zhang, X., Cui, L., Wu, Z., Chen, K., Wang, D., and Liu, Q. H. (2020), "Imaging Hydraulic Fractures Under Energized Steel Casing by Convolutional Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing.*

Zhang, X., Henao, R., Gan, Z., Li, Y., and Carin, L. (2018), "Multi-label learning from medical plain text with convolutional residual models," *Machine Learning for Healthcare.*

Zhao, K., Rawat, V., Lee, S.-C., and Lee, J.-F. (2007), "A domain decomposition method with nonconformal meshes for finite periodic and semi-periodic structures," *IEEE Transactions on Antennas and Propagation*, 55, 2559–2570.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020), "A comprehensive survey on transfer learning," *Proceedings of the IEEE.*

Biography

Runren Zhang was born in Yongji, Shanxi Province, China. He received the B.S. degree in information and communication engineering in 2012 and the M.S. degree in electronic science and technology in 2015, respectively, both from Zhejiang University, Hangzhou, China. He completed the Ph.D. degree with the Department of Electrical and Computer Engineering at Duke University, Durham, NC, USA in August 2020. His current research interests include forward modeling with spectral element and discontinuous Galerkin methods, absorbing boundary conditions, and deep learning-based inverse problems.

Publications

1. Zhang, R., Sun, Q., Zhuang, M., Huang, W.-F., Zhan, Q., Wang, D., and Liu, Q. H. (2018), Optimization of the Periodic PML for SEM, *IEEE Transactions on Electromagnetic Compatibility*, 61, 1578–1585.

2. Zhang, R., Sun, Q., Wu, Z., Fang, Y., Hu, Y., Huang, W.-F., and Liu, Q. H. (2019), Fast Induction Logging Modeling With Hierarchical Sudoku Meshes Based on DGFD, *IEEE Geoscience and Remote Sensing Letters*, 16, 1683–1687.

3. Zhang, R., Wu, Z., Sun, Q., Zhuang, M., Cai, Q.-M., Wang, D., and Liu, Q. H. (2019), Memory-efficient 3-D LWD solver with the flipped total field/scattered field-based DGFD method, *IEEE Geoscience and Remote Sensing Letters.*

4. Zhang, R., Sun, Q., Zhang, X., Cui, L., Wu, Z., Chen, K., Wang, D., and Liu, Q. H. (2020), Imaging Hydraulic Fractures Under Energized Steel Casing by Convolutional Neural Networks, *IEEE Transactions on Geoscience and Remote Sensing.*