A Novel Approach to Reservoir Simulation Using Supervised Learning

Shahdad Ghassemzadeh, B.Sc. (Hons), M.Sc.

A thesis submitted for the degree of

Doctor of Philosophy (Ph.D.)

Australian School of Petroleum and Energy Resources
Faculty of Engineering, Computer & Mathematical Sciences
The University of Adelaide

November 2020

Dedication

This dissertation is lovingly dedicated:

To my mother, Shahla, and my father, Alireza, for their eternal support, endless encouragement, and constant love, which have sustained me throughout my life.

In memory of my grandmother, Shahrbanoo and grandfather Parviz. Although they are no longer of this world, their memories continue to regulate my life.

Table of Contents

List of Figures

List of Tables

Abstract

Numerical reservoir simulation has been a fundamental tool in field development and planning. It has been used to replicate reservoir performance and study the effects of different field conditions in various reservoir management scenarios, and during field development and planning. Consequently, physics-based simulations have been heavily used during various reservoir studies such as history matching, uncertainty quantification and production optimisation; grid size and geological complexity also have a significant influence on the speed of the simulation. Furthermore, heterogeneities such as natural or hydraulic fractures can cause convergence problems and make the simulation even more time-consuming and computationally expensive. Due to being computationally demanding, such studies are also extremely time intensive. As a result of this downside, it is practically impossible to follow workflows such as the closed-loop reservoir management approach, which recommends updating the model every time a set of new data is available.

Additionally, any management scenario must be approached from a business and economic standpoint. This means that, based on the predefined objectives within the study, the respective layers of precision must be chosen by the user. Therefore, if less expensive techniques can be implemented and provide adequate results, the use of more accurate and costly methods cannot be justified.

One popular solution in overcoming this problem involves the creation of an approximate proxy model for the required features of the desired reservoir. This is achieved by either replacing or combining the physics-based model with this approximate model. However, by following this approach, the designed proxy model is only able to represent its corresponding reservoir, with a new proxy model needed to be rebuilt from scratch for any new reservoir. With consideration to the overall runtime, it can be observed that the time taken in iteratively running a numerical reservoir simulation may be faster than the time taken by the entire process

of building, validating and using a proxy model. Therefore, this thesis focuses on the feasibility, advantages and contribution of a complete stand-alone AI-based simulator, Deep Net Simulator (DNS), in a wide range of different conventional and tight sand reservoir scenarios in 1D, 2D and 3D space.

This thesis involves the use of deep learning to create a data-driven simulator, Deep Net Simulator (DNS), that enables the simulation of a wide range of reservoirs. Unlike conventional proxy approaches, a large amount of data is collected from multiple reservoirs with varying configurations and complexities. This results in the creation of a comprehensive database, including various possible reservoirs' features and scenarios. The hypothesis is that such an approach will enable the data driven model to perceive and understand the principles that make up reservoir modelling and that the model will act as an excellent approximation to the equations that traditional physics-based numerical simulators solve. This objective is highly possible, since deep learning has been shown to be a great universal function estimator, which is capable of estimating the physics once given enough data and observations. Hence, this thesis aims to develop a series of data-driven models with the aforementioned features for various types of reservoirs.

Initially, a workflow is designed to integrate a commercial simulator with a data extraction algorithm, enabling the generation of input-output simulation datasets. Next, the datasets are generated and reviewed. These datasets are then used in the training, validating and testing of the developed models. These developed data-driven models are able to learn and reproduce the physics governing fluid flow for a range of different scenarios: a single-phase oil reservoir in one-dimensional space, a single-phase gas reservoir in two-dimensional space, a single-phase gas reservoir in three-dimensional space, and hydraulically fractured tight gas reservoirs in two-dimensional space.

The developed model was evaluated in terms of precision, speed, and reliability. For each scenario, the developed model was compared with a commercial reservoir simulator, and its performance was assessed using the following metrics: mean absolute error, mean absolute percentage error (MAPE), mean relative error, mean square error, root mean square error and r squared. The developed model was able to predict 45%, 70% and 90% of the cases with less than 5%, 10% and 15% MAPE, respectively. Furthermore, depending on the number of cells requiring outputs, the developed model was able to reduce runtime by 100% up to 1.04E+08%.

This thesis takes the first steps towards establishing a new approach using AI and deep learning, for reservoir management procedure that is cheaper, less computationally demanding and more adaptable. This approach may result in a better value creation alongside a quicker decision-making process and, possibly, the advantage of integrating other attributes and data that are currently not used in physics-based models.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and, where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Acknowledgement

My findings and development as a researcher would not have been possible without those who supported me throughout the tenure of this work. First and foremost, I would like to express my sincere gratitude to Mrs Maria Gonzalez Perdomo, whose patient but firm guidance, as well as her endless support, even long before I started my PhD program, have been invaluable to my career growth and the accomplishment of this research. Her insightful guidance as both an academic and personal mentor and her kindness were truly exceptional and are much appreciated. Secondly, I express my deepest gratitude to Dr Manouchehr Haghighi, who exceeded any expectations I would have had from a co-supervisor. This project was possible because of his visionary guidance, ongoing encouragement to question my results, and being always available for technical suggestions. I am also grateful to Dr Ehsan Abbasnejad for committing his time and providing valuable feedback and recommendations on my manuscripts. Santos Ltd. partially provided financial support for me and this project, and this support is also much appreciated.

I would like to especially thank my mentor, Mr Teof Rodrigues, for believing in my potential and supporting me throughout the program and thank him for all the help and advice he has offered me. Many thanks to Dr Abbas Zeinijahromi, Dr Alireza Salmachi, Dr Kathryn Amos, Dr Pavel Bedrikovetski, Dr Peter McCabe and Dr Simon Holford and other faculty members for always providing help and support throughout my stay in the Australian School of Petroleum & Energy Resources.

I would also like to express my sincere appreciation to Dr Behnam Ferdosi, Dr Mahin Schaffie, Dr Mansour Kalbasi, Dr Mohammad Ranjbar, Dr Mohammadjafar Mohammadzadeh, Dr Peyman Pourafshary and Dr Yashar Pourrahimian, my former professors, whose guidance and support enabled me to be admitted to this postgraduate program.

As for friends, I would like to sincerely thank Dr Mani Moujerloo, (soon to be) Dr Amir Arya, Dr Amir Hashempour Charkhi, Dr Farzin Ghanadi, Dr Nima Sedaghatizadeh and Dr Shaghayegh Dezvarei for everything they have given me over the last nearly four years, being compassionate, thoughtful and intelligent friends and conversationalists, giving me many new perspectives and making my time in Adelaide and Australia much more fun, while also helping to keep me grounded. I would also like to thank Roozbeh Koochak, Dr Gabriel Malgaresi and Yazan Arouri, who were always eager to assist me and project their positive energies into my life. Moreover, I thank my departmental colleagues and friends for enthusiastically listening to me talk about my findings. I want to express my deep appreciation for Alexander Arthurson's help throughout the process of writing this thesis and I'd also like to extend my sincere gratitude to Dr Alison-Jane Hunter for taking the time to proofread the entire thesis meticulously.

Most of all, I am immensely thankful to my family for loving me unconditionally, supporting me in every second of my life and in everything I do, and continuously sacrificing themselves to see me succeed: to my parents, Shahla and Alireza, my heroes and the most important people in my life, who taught me to preserve my integrity and honesty, no matter what it takes, to keep dreaming big and encouraged me to achieve my goals; to my elder brother, Shobeir, who always inspired me with his work ethic and diligence; to my second elder brother, Shantia, who always inspired me with his endless kindness and ambition; to my sisters-in-law, Mahjoobeh and Zohre, who have been a constant source of emotional support and encouragement; to my beloved and adored nephews, Amir Arad and Barsin, for whom my love and longing are beyond words; to Sadaf and Mehrdad, who hold a special place in my heart and last but not least to my aunt, Farideh, who has also been generous with her love. Thank you all for being my soundboards, my most generous supporters, and my biggest distractions. Without such a team behind me, I doubt that I would be in this place today. I hope that I have made you proud.

List of Publications

Journal Papers

- Ghassemzadeh, Shahdad; Gonzalez Perdomo, Maria; Haghighi, Manouchehr; Ehsan Abbasnejad. "Development of a Data-Driven Model for the Real-Time Forecasting of Natural Gas Reservoirs' Behaviour " (Not submitted).
- Ghassemzadeh, Shahdad; Gonzalez Perdomo, Maria; Haghighi, Manouchehr; Ehsan Abbasnejad. "Deep Net Simulator (DNS); Implementing a Fast and Reliable Data-Driven Reservoir Simulation for Natural Gas Reservoirs" Journal of Neural Computing and Applications (Conditionally accepted).
- Ghassemzadeh, Shahdad; Gonzalez Perdomo, Maria; Haghighi, Manouchehr; Ehsan Abbasnejad. "Deep Net Simulator (DNS): a new insight into Conventional Gas Reservoir Simulation, APPEA Journal, 60,124-132., 2020.

Conference Papers:

- Ghassemzadeh, Shahdad; Gonzalez Perdomo, Maria; Haghighi, Manouchehr; Ehsan Abbasnejad. "Modelling Hydraulically Fractured Tight Gas Reservoirs with an Artificial Intelligence (AI)-based simulator, Deep Net Simulator (DNS), EAGE Digitalization Conference and Exhibition; Vienna, Austria; Dec. 2020.
- Ghassemzadeh, Shahdad; Gonzalez Perdomo, Maria; Haghighi, Manouchehr." Application of Deep Learning in Reservoir Simulation", EAGE Petroleum Geostatistics; Florence, Italy; Sep. 2019.

List of Abbreviations

Adagrad	adaptive gradient algorithm
ANN	artificial neural network
BHP	bottom-hole pressure
BPNN	back-propagation neural network
CNN	convolutional neural network
DNS	deep net simulator
ELU	exponential linear unit
MAE	mean absolute error
MAPE	mean absolute percentage error
MRE	mean relative error
MSE	mean squared error
PRE	percentage relative error
R2	R squared, correlation coefficient
RBNN	radial basis neural network
RE	relative error
ReLU	rectified linear unit
RMSE	root mean squared error
SAGD	steam-assisted gravity drainage
SGD	stochastic gradient descent

List of Symbols

A	drainage area, sq ft
B_g	gas formation volume factor (FVF) (RB/SCF)
B_o	oil formation volume factor (RB/STB)
B_w	water formation volume factor (RB/STB)
g	activating function
h	hypotheses
k_r	relative permeability, md
q_g	gas flow rate, Mscf/D
q_o	oil flow rate, STB/D
R_s	dissolved GOR, scf/STB
S	saturation (fraction)
w	Weights of a neural networks
X	input vector
α	learning rate
β	turbulence factor
η	learning rate
θ	Parameters of a neural networks: weights and biases
μ	viscosity, cp
ρ	density, lbm/ft^3
Υ	gravity
Υ	momentum term
ϕ	porosity, dimensionless

1. Introduction

1.1.Research Background and Rationale

Reservoir Simulation is a fundamental concept for achieving both a deep understanding and a detailed interpretation of a reservoir's behaviour. This is crucial in studies involving reservoir management and field development planning within the oil and gas industry. The reservoir's behaviour is quantified by using a series of partial differential equations called flow equations. These equations have been derived from the integration of three physical concepts: mass conservation, thermodynamical equilibrium and fluid flow (and may or may not include heat transfer). Using numerical methods such as the finite-difference method, these equations can be solved throughout the entire reservoir. This outcome is achieved through the use of high computational power, allowing for the prediction of flow through porous media; therefore, this prediction enables the interpretation of the performance of a reservoir over a range of different development scenarios. By incorporating these profiles into their respective oil and gas price forecasts, industries are then able to generate cash flow predictions, later to be used in quantifying the financial outcome for any given scenario.

Currently, depending on if the required rock, fluid and production data, are available, physics-based numerical simulation is the most popular method for predicting the reservoir's performance. Reservoir simulations are typically used for the following outcomes: to quantify reservoir uncertainties, calibrate the simulation model through history matching, and to optimise the production of the desired field. Each of these respective tasks would generally require the completion of hundreds, or even thousands, of simulations to examine various possible scenarios. Although the introduction of powerful hardware and software in recent years has improved the reservoir models and simulations indisputably, computational inefficiencies and slow runtimes are still ongoing issues within some reservoir simulation areas, such as history matching and optimisation applications. The main reason for this slow runtime is linked to the expansion of the static models' sizes in recent years, and the complex physics needing to be taken into account. A static model integrates all the collected features into one single network and then represents the entire reservoir as interconnected blocks in space with its associated attributes. Given that more advanced technologies in data collection have been introduced in recent years, considerably more data are available, resulting in the same interconnected network being represented by many more cells. In addition to the slow nature of reservoir simulation itself, the number of scenarios that must be examined in order to find the optimum configuration has caused a significant annual price rise for the computational expenses of reservoir optimisation and management. Although this runtime can be partially reduced by using high-speed computers, there is still a considerable associated cost in doing this, justifying the investigation of alternative approaches. Hence, designing and implementing faster reservoir simulators is one of the main priorities within the oil and gas industry.

In the recent decade, the expansion of the implementation of automation within varying operations and services has boomed within the petroleum industry. This expansion has led to the development of smart field technologies, allowing for the almost continuous collection of field data. However, the collection of this massive amount of data causes a new challenge as to how to best deal with this data to create useful values and thus make better decisions. The closed-loop reservoir management approach is recommended as an efficient approach to overcome the challenges faced when a massive amount of data is present, see Figure 1.1 (Jansen, Brouwer et al. 2009). This approach involves updating models continuously, as soon as new information is available. When using this workflow in decision-making, it is crucial that the model can fulfil the following criteria:

- Update quickly once new data is presented.
- Be accurate enough to represent the actual system.

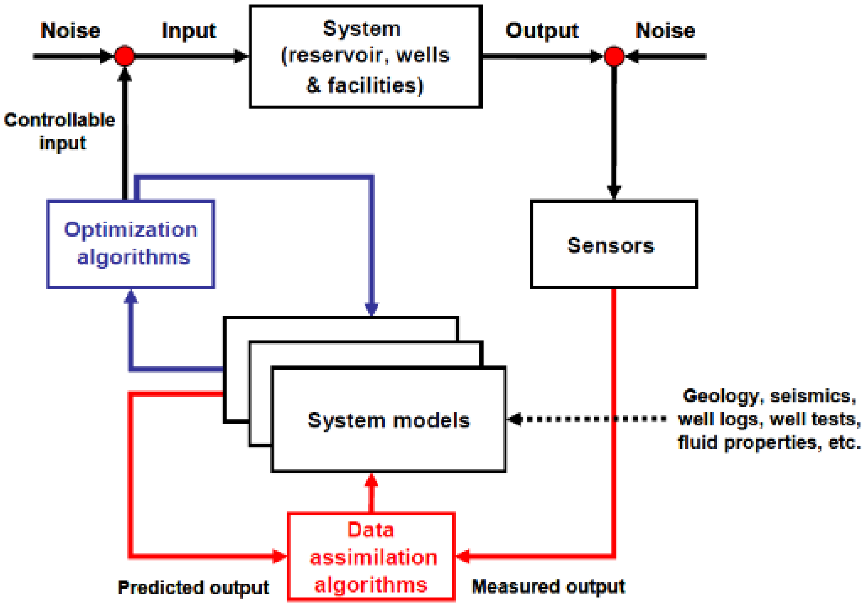

Noise
Input
System (reservoir, wells & facilities)
Output
Noise
Controllable input
Optimization algorithms
Sensors
System models
Geology, seismics, well logs, well tests, fluid properties, etc.
Data assimilation algorithms
Predicted output
Measured output

Figure 1.1, A scheme of the closed-loop reservoir management approach (Jansen, Brouwer et al. 2009)

Such a workflow could potentially be obtained through reliance on AI techniques. These techniques could enable the development of data-driven models, capable of reproducing the relationship existing between the input and output variables during the training phase. The key difference between data-driven and physics-based models is the presence of an accepted visible correlation, used to describe the natural phenomena in physics-based models. Unlike physics-based models, data-driven models do not assume such tangible relationships and instead capture the phenomenon's information via the extraction of features that existed within the observed data.

One of the most popular applications of data-driven models in reservoir engineering is the creation of a proxy model with the desired outputs as a replica of the simulation model of a specific reservoir. Through the application of mathematics functions, the proxy model of a corresponding reservoir can approximate the response of that system quickly and without requiring a significant amount of computational simulations (Artun, Ertekin et al. 2011, Goodwin 2015, Kalantari-Dahaghi, Mohaghegh et al. 2015, Alenezi and Mohaghegh 2016, Ghassemzadeh and Charkhi 2016, He, Xie et al. 2016, Alenezi and Mohaghegh 2017, Chen, He et al. 2017, Mohaghegh, Gaskari et al. 2017, Kim and Shin 2018, Nwachukwu, Jeong et al. 2018). Model inputs usually include initial conditions, operational configurations, and reservoir characteristics, such as porosity, permeability, etc. Model outputs include production or saturation profiles and recovery facts among other variables.

1.2.Research Gap

While these methods have provided a significant step forward in the field, each of these proxy models can be used only for their corresponding reservoirs, since they are Specifically built based on data from these reservoirs. This means that for each new reservoir, a new proxy model must be constructed from scratch. Furthermore, if the time spent building, validating, and running a proxy model is considered, it may be observed that running a physics-based numerical reservoir simulator iteratively could result in comparatively greater time efficiency. Therefore, seeking a faster alternative approach is inevitable.

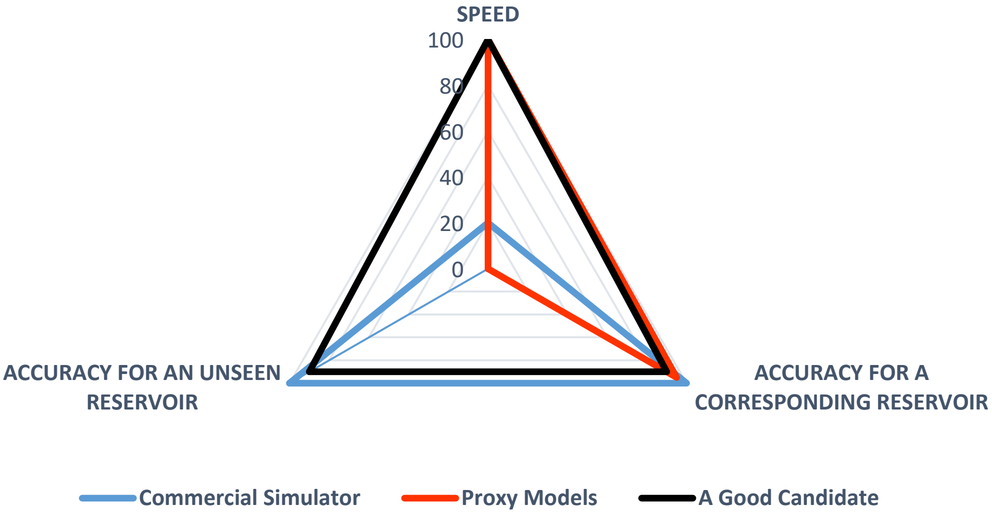


Figure 1.2, A rough comparison of accuracy and speed of a commercial simulator, a proxy model and a possible suitable replacement.

As roughly illustrated in Figure 1.2, to be considered a faster alternative, the right candidate must satisfy both prerequisites of the closed-loop reservoir management approach. In other words, it must have both the accuracy and generalisation of a commercial simulator and the runtime of a proxy model. (Ghassemzadeh, Perdomo et al. 2019).

1.3.Research Aim and Objectives

The aim of this research is to investigate the possibility of *developing a fast and reliable alternative simulator: a data-driven simulator.*

This model is designed to be a forecasting tool, to instantly predict the pressure at any point within the reservoir once given the initial conditions, operational parameters, and reservoir characteristics. An advantage of using such simulator is that it enables engineers to investigate a wide range of scenarios in a shorter timeframe and thus present a more attractive approach. This approach should be interpreted as a data-driven clone to a numerical reservoir simulation, as it represents multiple reservoirs in place of the single reservoir approach dictated by conventional proxy modelling. Therefore, the outcome of this research can be considered a data-driven simulator. Ideally, we should learn directly from field observations, as this allows us to reduce the uncertainty associated with the fluid flow equations. However, since this research is a feasibility study, we used a commercial simulator to generate the required database as the surrogate to field observations in this thesis. When completed, this data-driven simulator is pre-trained and independent of any commercial simulators.

This investigation will be followed by the development of a prototype of a simulator for a series of reservoir types and conditions. This data-driven simulator could be a game-changer in the industry, as it provides a fast and reliable means to achieve every decision-making stage. A better decision-making workflow will result in improvement in the efficiency of the reservoir management and development planning studies, leading to an increased profit margin and greater productivity in the long run. The following objectives are studied in this research:

- To design a workflow that integrates a commercial simulator with a data-mining algorithm to generate input-output simulation datasets (labelled data)
- To generate labelled data for a single-phase one-dimension oil reservoir, and build and train a data-driven model to the extent that it can learn and reproduce the physics governing fluid flow for these assumed conditions
- To produce labelled data corresponding to two-dimensional natural gas reservoirs and build and train a data-driven model and investigate its accuracy using multiple scenarios with unseen features and characteristics
- To build and develop a data-driven simulator for hydraulically fractured tight gas reservoirs in the two-dimension space
- To investigate the practicality of the proposed workflow in creating a data-driven simulator for natural gas reservoirs in the three-dimension space
- To validate the developed data-driven simulator using real field data

1.4.Thesis Structure

This thesis consists of three principal sections: the literature review, feasibility study and model construction.

In the first part (Chapter 2), a brief review of reservoir simulation and deep learning are presented. In addition to these, a comprehensive literature review of previous applications of machine learning, specifically for artificial neural networks, in reservoir simulation is presented.

Next, in the second part (Chapter 3), a feasiblility study for using a data-driven model as an independent simulator is presented. In this study, the possibility of replacing fluid flow equations with a data-driven network is explored. Therefore, as a part of this study, a commercial simulator coupled with a data-mining algorithm will be generated, resulting in the development of a workflow that generates input-output data sets for any desired scenarios. Two data-driven simulators were developed: one for one-dimensional oil reservoirs and one for two-dimensional gas reservoirs. Both developed simulators used the same inputs required by a physics-based simulator and are able to predict the pressure drop throughout the reservoir. During the feasibility study, a sensitivity analysis was used to understand the effect of different input parameters. The proposed model presented a novel approach for modelling fluid flow in a porous media, based on fluid and rock properties alongside production characteristics. The study presents a fast approach, leading to exploration of more scenarios in history matching and optimization. Having explored more scenarios, engineers can provide better future cash flow predictions.

In the final part (Chapters 4, 5 and 6) the designed workflow was improved using the feedback of the first two scenarios, allowing a bigger dataset to be created. The use of big data results in a data-driven network with higher capacity and, consequently, a more accurate model. This modified workflow was used to develop several versions of the developed data-driven simulator for conventional and unconventional gas reservoirs in two and three dimensions, as elaborated below.

The first study was shaped around a data-driven model of a gas reservoir in a 2-dimensional space. A wide-ranging dataset was generated using a data-mining algorithm. This dataset was pre-processed, then fed into a deep learning model. The model was trained and validated via 80% of the generated data. Next, the trained model was tested by the unseen test data, the remaining 20% of the generated dataset. Then, in order to examine the accuracy of the model even further, the developed model was tested through multiple scenarios for various input parameters: the initial reservoir pressure, porosity and permeability quantity and distribution, well location, well rate and bottom hole pressure. Each scenario was a complete synthetic reservoir model, and we were able to compare the outputs of the developed model with a commercial simulator. In total, 53 scenarios, consisting of around 8,000,000 data points were used to confirm the accuracy of the model and, consequently, the developed approach.

The next study was focused around upgrading the 2-dimensional model to a 3- dimensional one, in order to make the developed model more practical. Furthermore, three different PVT datasets were used, as opposed to only one set in the 2D model. As the 3D fluid dynamic is more complicated, we needed a bigger dataset. Therefore, in the development of this version of the model, 40,000,000 data points, corresponding to 100 synthetic models, were used. A similar data-mining workflow was implemented, and then the generated data set was fed to a commercial simulator. Similar to the previous case, the model was verified by unseen test sets and multiple unseen scenarios. To confirm the accuracy of the developed model, 600 scenarios, which consisted of more than 500,000,000 data points, were simulated by both the developed model and a commercial simulator. In addition to examining the accuracy, the model’s speed was also compared with that of a commercial simulator.

The final study was focused on the development of a data-driven simulator for tight gas reservoirs, under hydraulic fracture operations. This was a challenging scenario, as the transmissibility is altered significantly in the fractures and the porous media in this type of reservoir. Again, the developed model was examined using unseen test sets and 140 benchmarks.

1.5.Chapter Overview

This thesis is written based on the thesis by publication format. In total, five publications are included in this thesis; one article has been published in peer-reviewed journals, two extended abstracts have been published in distinguished conferences, one article has been conditionally accepted, pending final feedback from reviewers and one article is currently pending submission.

Table 1.1, List of publications and associated chapters.

Paper	Chapter	Title	Status
1	3	A novel approach for Dynamic Reservoir simulation using Deep Learning Network	published
2	3	Deep Net Simulator (DNS): a New Insight into Reservoir Simulation	published
3	4	Development of a Data-Driven Model for the Real-Time Forecasting of Natural Gas Reservoirs' Behaviour	Not submitted
4	5	Deep Net Simulator (DNS); Implementing a Fast and Reliable Data-Driven Reservoir Simulation for Natural Gas Reservoirs	Conditionally accepted
5	6	Modelling Hydraulically Fractured Tight Gas Reservoirs with an AI-based simulator, Deep Net Simulator (DNS)	published

The body of this thesis is formed by seven chapters. The first chapter includes an introduction to the significance of the research, general aim and objectives and structure of the dissertation. The second chapter presents a brief introduction to reservoir simulation and deep learning. This chapter also includes a comprehensive literature review of the research background in the application of machine learning to reservoir simulation. The third chapter presents a feasibility study in the development of a data-driven simulator. The fourth chapter presents the development of the two-dimension data-driven simulator and reports its accuracy. The fifth chapter presents a complete study of the accuracy and speed of the developed data-driven simulator in a three-dimension space. The sixth chapter presents the development of another version of the developed model for hydraulically fractured tight gas reservoirs. The seventh chapter summarises the research work in this thesis and presents recommendations for future work.

2. Literature Review

2.1.Reservoir Simulation

Oil and gas reservoirs are a set of lithological units that contain porous and permeable media. This porous media is formed of either sedimentary or carbonate layers, and hydrocarbons can be accumulated inside these pores. Stored fluids are trapped by an overlying impermeable layer, like shale. Figure 2.1 shows a simplified cross-section view of a typical hydrocarbon reservoir. To have a better understanding of the reservoir and make proper investment decisions, companies perform reservoir simulations. They regularly perform and tune these simulations throughout the life of a field from exploration to development, production and enhanced oil recovery.

The first step to simulate a reservoir is to gather geological data and create a static model. By discretising the reservoir, reservoir simulators subdivide a reservoir into a series of interconnected blocks (finite volume elements). Gathered data is then assigned to each block; Figure 2.2. Flow-equations are expressed in terms of partial differential equations written in a finite-difference form, which discretises the problem in time and space. This discretisation allows the simulator to compute the fluid flow through the entire reservoir. Each cell includes different variables, such as petrophysical attributes, which represent the characteristics of the reservoir. In the simplest case, the grids are cube-shaped cells. Due to the geological complexity of the internal structure of reservoirs, such as faults and fractures, an appropriate and acceptable reservoir model comprises millions of cells (Fanchi 2001).

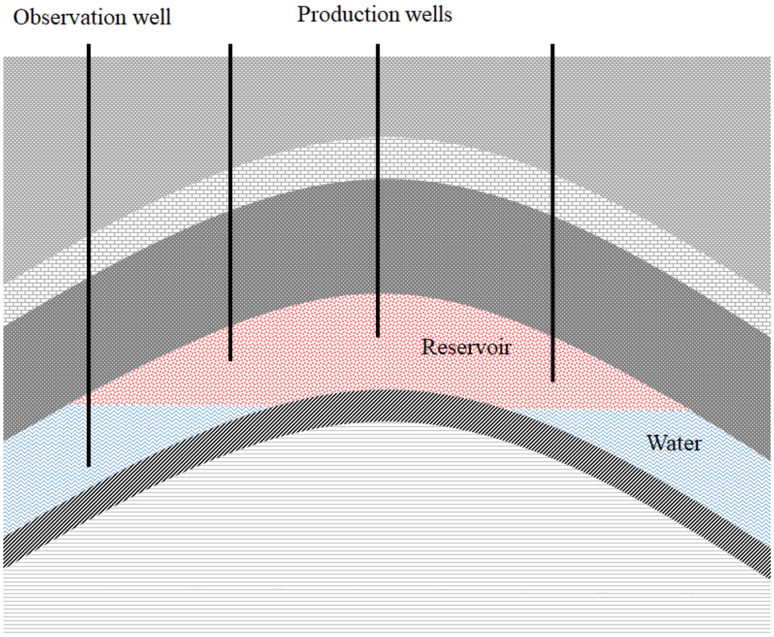


Figure 2.1, Cross-sectional scheme of a typical oil/gas reservoir.

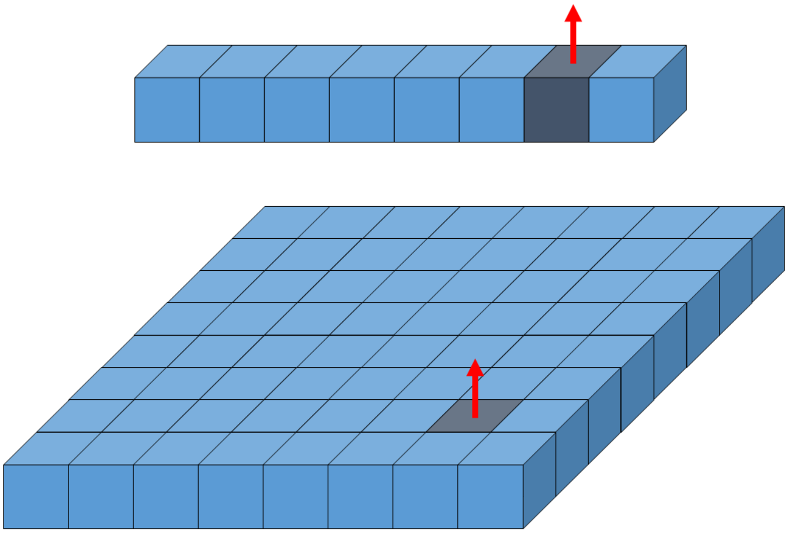


Figure 2.2, Schemes of grids for 1D (Top) and 2D (Bottom) reservoir simulation.

As the first step to simulate a reservoir and its fluid's behaviour, a static model must be generated using the available data. Well and seismic data are then used to create a geological model, containing fine cells of rock lithofacies, porosity, permeability, fluid saturation, and faults location. Then, all the properties of this model are up-scaled to a reservoir model, which still can have up to millions of cells. After assigning boundaries and initial conditions based on the type of the reservoir, the simulator solves the following or similar fluid flow equations for each time step and cell:

$$\frac{\partial}{\partial x}\left[\beta_c k_x A_x \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial Z}{\partial x}\right)\right]\Delta x + \frac{\partial}{\partial y}\left[\beta_c k_y A_y \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial Z}{\partial y}\right)\right]\Delta y + \frac{\partial}{\partial z}\left[\beta_c k_z A_z \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial z} - \gamma_o \frac{\partial Z}{\partial z}\right)\right]\Delta z = \frac{V_b}{\alpha_c}\frac{\partial}{\partial t}\left(\frac{\varphi S_o}{B_o}\right) - q_{osc} \tag{1}$$

$$\frac{\partial}{\partial x}\left[\beta_c k_x A_x \frac{k_{rw}}{\mu_w B_w}\left(\frac{\partial p_w}{\partial x} - \gamma_w \frac{\partial Z}{\partial x}\right)\right]\Delta x + \frac{\partial}{\partial y}\left[\beta_c k_y A_y \frac{k_{rw}}{\mu_w B_w}\left(\frac{\partial p_w}{\partial y} - \gamma_w \frac{\partial Z}{\partial y}\right)\right]\Delta y + \frac{\partial}{\partial z}\left[\beta_c k_z A_z \frac{k_{rw}}{\mu_w B_w}\left(\frac{\partial p_w}{\partial z} - \gamma_w \frac{\partial Z}{\partial z}\right)\right]\Delta z = \frac{V_b}{\alpha_c}\frac{\partial}{\partial t}\left(\frac{\varphi S_w}{B_w}\right) - q_{wsc} \tag{2}$$

$$\frac{\partial}{\partial x}\left[\beta_c k_x A_x \frac{k_{rg}}{\mu_g B_g}\left(\frac{\partial p_g}{\partial x} - \gamma_g \frac{\partial Z}{\partial x}\right) + \beta_c k_x A_x \frac{k_{ro} R_s}{\mu_o B_o}\left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial Z}{\partial x}\right)\right]\Delta x + \frac{\partial}{\partial y}\left[\beta_c k_y A_y \frac{k_{rg}}{\mu_g B_g}\left(\frac{\partial p_g}{\partial y} - \gamma_g \frac{\partial Z}{\partial y}\right) + \beta_c k_y A_y \frac{k_{ro} R_s}{\mu_o B_o}\left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial Z}{\partial y}\right)\right]\Delta y + \frac{\partial}{\partial z}\left[\beta_c k_z A_z \frac{k_{rg}}{\mu_g B_g}\left(\frac{\partial p_g}{\partial z} - \gamma_g \frac{\partial Z}{\partial z}\right) + \beta_c k_z A_z \frac{k_{ro} R_s}{\mu_o B_o}\left(\frac{\partial p_o}{\partial z} - \gamma_o \frac{\partial Z}{\partial z}\right)\right]\Delta z = \frac{V_b}{\alpha_c}\frac{\partial}{\partial t}\left(\frac{\varphi S_g}{B_g} + \frac{\varphi R_s S_o}{B_o}\right) - q_{gsc} \tag{3}$$

These equations represent the flow of oil, water, and gas in porous media in three dimensions, respectively. These flow equations contain all the fundamental principles we are dealing with in reservoir engineering: mass conservation, thermodynamic equilibrium and fluid flow. Using the finite-difference method, we discretise these partial differential equations in time and space and make these equations computable throughout the entire reservoir (Fanchi 2001, Abou-Kassem, Farouq-Ali et al. 2013).

Given there are no analytical solutions available for such complex equations, numerical methods are used to solve them. To avoid any convergence problem in the solving procedure, engineers mostly use the fully implicit finite difference method. Based on the field conditions and desired economic outcome, parameters such as reservoir and fluid characteristics, well and surface facility configuration, and geological condition play a vital role in the final solution.

2.2.Proxy Modelling

Proxy modelling, as well as response surface methods, are used to discover and analyse the relationship that is found between explanatory variables with a particular response variable. The sections below discuss the various proxy models and how to use them best in a practical application.

2.2.1. Proxy Model Types

Through application of the simple mathematics function of the input parameters, proxy models approximate the response of a system without requiring computationally-demanding mathematical modelling. In the reservoir simulation context, proxy models can be used to predict model behaviour instantly and to quantify the change of an input value on output. These are significant advantages, as typically each of these tasks would be time and computationally intensive.

Proxy models are typically designed and built based on the results from multiple simulations. Statistical methods are usually used to select the minimum amount of simulated runs so as to obtain the maximum amount of information based on the uncertainty space (Friedmann, Chawathe et al. 2001, Yeten, Castellini et al. 2005, Montgomery, Peck et al. 2012)

There are several types of proxies that are generally used. Some of these proxies are heavily based on simple analytic functions, while others rely on the use of numerical approximations that cannot be represented clearly through simple functions.

- Analytic proxy

Analytic proxy is done through the analysis of an analytic function such as polynomial functions. This proxy is designed and built by fitting a suitable mathematical function to the respective data points through the use of regression techniques such as the least-squares technique. These techniques result in the determination of the coefficients required to minimise the cost function between the real data points and the fitted numerical function.

- Numerical proxy

Like the Analytic proxy, the Numerical Proxy is used to approximate the relationship between the input factors against the respective output values. However, in numerical proxies, this relationship is obtained through a response surface developed using a numerical algorithm as opposed to an analytical function. Three of the most popular numerical proxy methods consist of Kriging, Splines and neural networks.

Kriging is used to predict the value of a function at a sampling point through the calculation of the weighted average of the function values, based on the data points. Kriging assumes that all points are spatially correlated; this correlation is described by a variogram model, which is a function of the Euclidean distance between two points. The larger the gap between these points, the more variant these two points are. The weights generated in computing the weighted average are obtained from the covariance from the sampling point, each data point respectively, and between all respective data points. If a data point is closer to the sampling point, a larger weight is assigned to this point. These weights are computed in such a way that the cost function of the fitted function is minimised (Journel and Huijbregts 1978, Deutsch and Journel 1992).

Spline function is a form of interpolation that is defined by a special type of piecewise polynomial. At the polynomial's connection, the point where its function value is known, the spline function has a significant level of smoothness (Li and Friedmann 2005).

Artificial Neural Network (ANN) is a popular machine learning method that has been used significantly in the creation of proxy models for reservoir simulation. The ANN method is inspired by the brain system in animals. ANN models consist of the interconnection of nodes, such as that of neurons in a brain. Each node accepts the inputs from other nodes and respectively generates outputs based on the inputs received and the node's internal stored information.

These nodes are systematically grouped into one or multiple layers. This input layer corresponds to receiving data from outside, and the data is then processed through the hidden layers in order to produce the corresponding response, which is called the output layer, Figure 2.3. The ANN needs to be trained to deliver the proper output response per given input accurately. This training is completed using training data where a known relationship is given between the output and input factors.

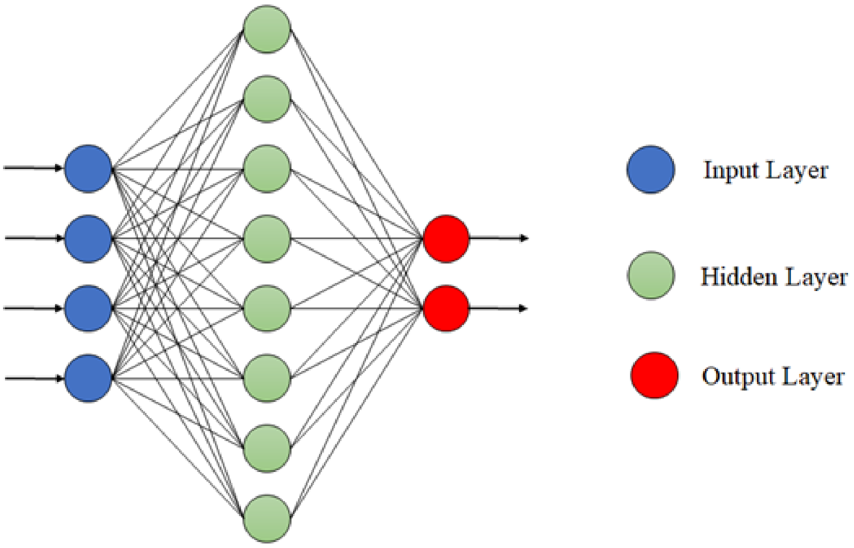


Figure 2.3, Neural networks schematic representation.

ANN is widely used within the petroleum industry in a range of different research areas including formation evaluation, reservoir thermodynamics, well placement and production optimisation. Furthermore, ANN has been used successfully to create proxy models for a range of reservoirs, such as: fractured reservoirs, shale gas reservoirs, conventional reservoirs, enhanced oil recovery and field optimisation.

Below, a brief review is presented for several successful implementations of proxy models for different purposes in the oil and gas industry:

Cullick et al. (2006) used a neural network to develop a proxy model for the history matching process. This model was used to minimise the difference between simulation results and field data. They applied the developed model in a water injection scenario. The reservoir model they studied consisted of 50,000 grid cells with approximately 141 million barrel oil in place. Data was generated from running a commercial simulation 50 times. The inputs of their model are permeability, porosity, net-to-gross and water contact depth, and the outputs are water injection rate and oil production rate. The developed ANN had a correlation coefficient of approximately 1. They achieved their goal in history matching through the fine-tuning reservoir and unknown parameters of wells, such as fluid production rate, pressure and water cut and using the developed proxy model. Even though they did not report how much faster their approach is, they concluded that such an approach could decrease the number of required simulation in the history matching process (Cullick, Johnson et al. 2006).

Mohaghegh et al. (2006) used fuzzy pattern recognition technology to develop a proxy model for a massive hydrocarbon field in the Middle East, consisting of close to one million grid blocks and more than 160 horizontal wells. Although the inputs of the model were not mentioned in the paper, the outputs of their model are the cumulative oil and water productions and water cut at any time for every horizontal well (Mohaghegh, Hafez et al. 2006)

Artun et al. (2011) developed an optimisation scheme in the cyclic pressure pulsing process with CO2 and N2 in naturally fractured reservoirs. They combined genetic algorithms and neural networks to achieve this. First, a series of proxy models were built with respect to the injected gas (CO2 or N2), and the reservoir fluid (heavy, volatile or black oils). To construct these proxy models, they fed neural networks data collected from a limited range of reservoir characteristics and design scenarios, with a correlation coefficient of 0.91 on average. Then, genetic algorithms were used to find the optimised operating conditions based on a cost function that considers cumulative injected gas and the time of the production. In the proxy model they made, they observed that the fracture permeability, ratio of fracture to matrix permeability, area, area-thickness product, oil-in-place, and oil saturation are the most critical parameters (Artun, Ertekin et al. 2011).

Dahaghi et al. used ANN to develop a proxy model for a shale reservoir model consisting of 5838 grid cells. This model was used to calculate the oil flow rate of a horizontal well as a function of 16 reservoir parameters, including matrix porosity and permeability, fracture porosity and permeability, hydraulic fracture properties, rock compaction and bottom hole pressure. The input data was generated by running the reservoir model 31 times for a 12-month production. Reporting an average correlation coefficient of 0.99, the authors reported that their model was fast and showed excellent accuracy when compared with a commercial simulator (Kalantari Dahaghi, Esmaili et al. 2012).

Memon et al. used a radial basis neural network (RBNN) and a back-propagation neural network (BPNN) to construct two proxy models. Their case study was a 3-layer synthetic reservoir with an initial pressure of 4800 psi. This reservoir included a gas injection well and an oil production well. Their models had 13 inputs including porosity, permeability, initial oil and water saturation. The developed neural models contained one hidden layer. The developed model could predict the bottom hole flowing pressure throughout ten years of production. When compared with a commercial simulator, the authors reported that RBNN and BPNN models showed an accuracy of 99.78% and 44.74%, respectively. They suggested that RBNN have more potential in comparison with BPNN in their case study (Memon, Yong et al. 2014).

Kalantari-Dahaghi et al. (2015a) proposed an ANN-based proxy model to estimate the production profile of a significantly hydraulic-fractured shale gas reservoir. The final model had one hidden layer with 55 neurons. The data set was generated with a commercial simulator and consisted of 98020 data points, which were used for training, validation and testing their model. The input of the developed model consists of: matrix porosity, matrix permeability, natural fracture porosity, natural fracture permeability, shale factor, hydraulic fracture height, hydraulic fracture length, hydraulic fracture conductivity, rock density, net to gross ratio, Langmuir volume, Langmuir pressure, diffusion coefficient, sorption time, and initial reservoir pressure. The developed model had a correlation coefficient of 0.99 and an error of less than 15% on average for the test sets when compared with a commercial simulator. They concluded that such an approach could be beneficial in reservoir management study and history matching (Kalantari-Dahaghi, Mohaghegh et al. 2015).

Later, these same researchers (Kalantari-Dahaghi et al., 2015b) used the same approach to replicate the production and injection profiles for the CO2-enhanced gas recovery and storage process.

They applied this method in a hydraulically-fractured Marcellus shale gas reservoir, located in Southwestern, Pennsylvania. The corresponding model for this reservoir was a three-layer model with 200,000 grid cells. To generate the spatio-temporal database, they simulated the reservoir model 20 times for 100 years and created a 116,000-point data set. The input data was similar to their previous model, but also included CO2 and CH4 properties. When compared with a commercial simulator, the model showed a correlation coefficient of 0.99 on average (Kalantari-Dahaghi, Mohaghegh et al. 2015).

Ghassemzadeh and Hasshempour introduced a two-proxy model in order to integrate a well production simulation software and reservoir simulation software. By using two different proxy models for the well and reservoir models, the new integrated model was more than just a simple approximation workflow. Each of the proposed proxy models has the ability to act as independent simulation software, and this approach provides a more reliable and flexible solution with respect to the whole system proxy. In other words, the proxy models learn the physics of the problem (reservoir and well) instead of being a simple copy of a specific reservoir and well. They integrated this workflow to a genetic algorithm optimiser to improve the net present value of the gas lift operation. They applied their model in order to perform real-time optimisation on a synthetic field with four wellbores. Their model showed a correlation coefficient of 0.99 and mean absolute percentage error (MAPE) of 2% on average, in comparison with commercial simulators. Using this approach, they enhanced the field production by 23%, when compared with the conventional approach (Ghassemzadeh, Charkhi et al. 2016).

Kulga et al. developed a data-driven model for tight gas reservoirs with hydraulic-fractured horizontal wells. Their model was designed to be used as a forecasting tool to predict the well behaviour of the horizontal well, once reservoir characteristic, operational configuration, the initial condition, and the hydraulic fracture parameters are given. This proxy was built using a shallow neural network with three hidden layers with 16,15 and 14 neurons. The developed model had 13 input variables: reservoir thickness, porosity, permeability, temperature, drainage area, length of the horizontal well, gas gravity, initial pressure, flowing bottom-hole pressure and the hydraulic-fracture properties. The outputs of this model were hyperbolic decline-curve coefficients. They tested their model with three unseen cases, in which well performance was forecasted with an average error of 4.9% and the cumulative gas recovery after ten years forecasted with an average error of 3.2% (Kulga, Artun et al. 2017).

Later, Kulga et al. (2017) continued this research on developing a data-driven model for tight gas reservoirs. They used a similar approach to determine both the reservoir characteristics and hydraulic fracture effectiveness, once provided with the operation parameters, initial conditions and the well performance characteristics. The developed ANN model consisted of three hidden layers with up to 57 hidden neurons. The outputs of this network were the reservoir characterisation and hydraulic fracture parameters, along with their respective probability distribution. They applied this model to an unseen case study and incorporated the observed production range into a Monte Carlo simulation. Once quantified, the probabilistic estimates of the parameters were found to be in agreement with the predefined uncertainties in the numerical model. The model was observed to have an average error ranging from 12.4% to 27.5%. Although the range is relatively high, it can be used as a quick fix for inverse problems due to its computational efficiency (Kulga, Artun et al. 2018).

Alenezi and Mohaghegh (2016) used ANN and data mining to build a proxy model for a reservoir, being water flooded with one injection well. This reservoir was located north-east of the Permian Basin in West Texas. The reservoir model consisted of 9,450,362 grid cells with more than two thousand wells drilled since 1948. Numerical simulation was used to generate a comprehensive spatiotemporal database. This database, then, was used to train, calibrate, and verify the developed model. The outputs of their proxy model were pressure and saturation for each cell. The inputs for the developed model were location, injection rate, depth, injection cumulative, thickness, production rate, porosity, production cumulative, permeability, pressure, distance to injection and boundaries, and saturation. Their model showed a correlation coefficient of approximately one in the training and test sets (Alenezi and Mohaghegh 2016).

Using the same approach, Alenezi and Mohaghegh (2017) created a proxy model for the northern platform of SACROC field. The result of this research was the development of a smart proxy model that can replicate simulation with 2% of MAPE (Alenezi and Mohaghegh 2017).

Shams et al. (2017) compared four popular methods used in creating proxy models: radial basis function, kriging, thin-plate spline, and ANN. Using each method, they developed four proxy models for the Gullfaks reservoir. The reservoir model for this reservoir consisted of 487,750 grid cells. The results showed that the ANN and kriging, with an average error of 28.88% and 25.38 % respectively, are more effective compared to radial basis function and thin-plate spline. The last two methods showed an average error of 38.09% and 35.85, respectively (Shams, El-Banbi et al. 2017).

Zheng et al. (2018) aimed to develop a proxy model to predict steam-assisted gravity drainage (SAGD) production profiles for arbitrary configurations in shale barriers. This model was achieved by training a shallow ANN with 21,600 data points gathered from 300 SAGD scenarios. The data they used in their study derived from a synthetic model collected from Suncor's Firebag project. The inputs of the model were: steam injection rate, elapsed time and shale location indexes; and the output is the oil production rate. It took the authors twelve hours to generate the training dataset, and around two minutes to train their ANN. The developed model generates output in less than three seconds. The authors tested the developed proxy model using 1008 data points from 14 unseen SAGD scenarios. The average MAPE for these 14 scenarios is 12.08% (Zheng, Leung et al. 2018).

Navratil et al. proposed a workflow to develop a proxy model of a reservoir capable of predicting the well production rate. In their approach, they consider the following features from the neighbourhood of the wellbore as inputs: rock type, porosity, permeability, pressure and saturation. In this workflow, they used a recurrent neural network arranged in an encoder-decoder architecture. They applied the proposed approach to the SPE9 benchmark model (Killough 1995). They reported that, using this approach, the simulation accelerated by 2000 times in well control optimisation. However, the accuracy of their model was quite high, as it ranges between 10% to 15% over a 20-40 month production period (Navratil, King et al. 2019).

Brantson et al. (2020) developed a proxy model to predict the recovery of a reservoir under low salinity water polymer flooding. They integrated particle swarm optimisation with an artificial neural network in the development of their model. The developed model consisted of three hidden layers with the following input parameters: saturation time, porosity, permeability, water saturation, polymer concentration, salt concentration and reservoir pressure. They used

3200 data points in the training of this model. Their model was able to show an accuracy of less than 1% of MAPE for the test set (Brantson, Ju et al. 2020).

As mentioned earlier, the downside to the conventional proxy models is that each is built based on data corresponding to a specific reservoir model. This means that it can only be used as a replacement for this corresponding model. Since a proxy model acts only as a replacement to its corresponding reservoir, in the case of a new reservoir, the entire process of building a proxy must be restarted. This includes redoing the entire time-consuming numerical simulation for the new reservoir in order to build the new proxy model. It implies that the limitation of this approach is that it always depends on the mathematical model.

2.3.Deep Learning

2.3.1. Introduction

A neural network is a name given to a sequence of many neurons, which are individually activated through either perceiving an environment or through the weighted connections from other neurons. It is possible to make the network learn, understand and exhibit the desired behaviour by assigning appropriate weights. The learning algorithm of a neural network can either be supervised or unsupervised. A neural network is considered supervised when the network learns to map an input to an output based on training sets with already-known outputs. A neural network that learns unsupervised has no such pre-existing labels. In unsupervised learning, the goal is to seek previously-undetected patterns among a training set with the least possible human supervision. Regardless of the type of the problem, a long combination of series of neurons and different activation functions may be required (Schmidhuber 2015).

Shallow neural networks, which typically comprise very few layers, have been around for many years. Backpropagation and gradient descent methods used in optimising the network, were developed in the 1960s. These methods have been applied to shallow neural networks since 1981; however, they were never successfully implemented into deep networks at that time. Nonetheless, after many years of vigorous research, supervised deep neural networks have earned many credits by winning many official pattern recognition competitions. (Schmidhuber 2015). Figure 2.4 shows a schematic of shallow and deep neural networks.

Deep Learning is a machine learning technique that can learn complicated relationships across data sets. It provides good approximations through the multiple layers with nonlinear transformation functions. In the learning process, functions' parameters and weights are optimised using the introduced data set. As mentioned, a large volume of data must be provided to the network to achieve optimised values for each parameter. This learning process extracts the useful features and information in hierarchical orders, allowing the model to obtain the complicated relationships of input data. It must be noted that features in higher levels are formed by the composition of features at lower levels (Schmidhuber 2015).

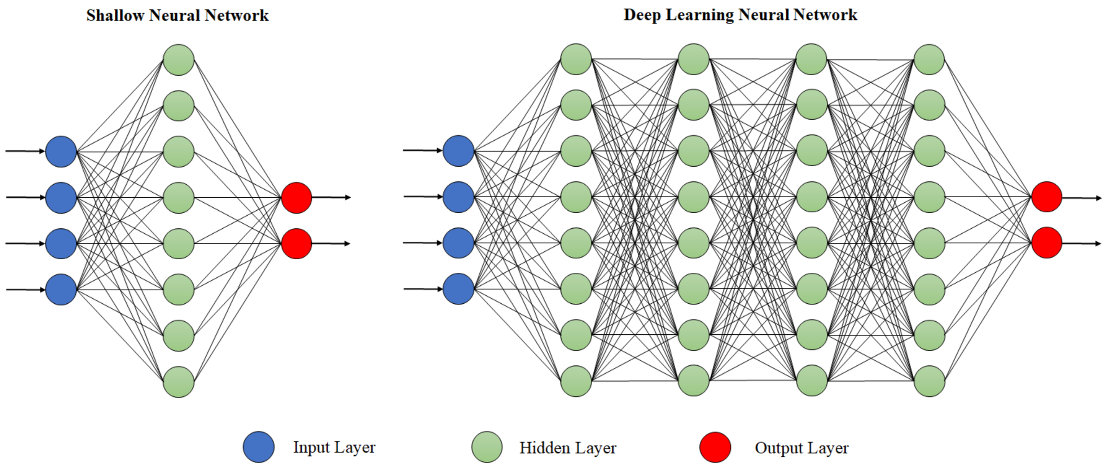


Figure 2.4, Shallow and deep neural networks: schematic representations

The depth of the architecture is one of the main differences that lies between standard artificial neural networks and deep learning networks. Additionally, more powerful optimisation functions are used in the learning process of deep learning networks. Given the recent progress in activation functions, optimisation algorithms, and hyperparameters, deep learning networks have enormous advantages over shallow neural networks (Schmidhuber 2015).

In deep learning, there are a series of layers between the input and output data; each identifying essential features of the data sets. Through this capability, the deep learning nets take away the time consuming, complicated, and sometimes even impossible feature selection task from the expert. Deep neural networks are exposed to raw data; next, they pre-process the data, extract and select critical features for complex mapping problems and use them for prediction or classification (Schmidhuber 2015, Goodfellow, Bengio et al. 2016).

Due to limited access to computational power and large data sets during the '80s and '90s, all experimental results were typically obtained from the few non-linear hidden layers in the traditional neural networks. It was not until 2006 that the computational power, data storage, and learning algorithms barriers were solved. Since then, deep learning nets have been successfully applied in many fields, including science (Mayr, Klambauer et al. 2016), biology (Alipanahi, Delong et al. 2015, Angermueller, Pärnamaa et al. 2016, Lore, Stoecklein et al. 2018), medicine (Giusti, Caccia et al. 2014, Das, Pradhapan et al. 2018), computer science and media (Hussain, Cambria et al. 2014, Barros, Jirak et al. 2015, Stettler and Francis 2018) and chemistry (Jeong and Lee 2018, Wu and Zhao 2018, Kim, Park et al. 2019).

Deep learning networks are truly an emerging technology gaining significant attention in image recognition, language translation, and signal/voice processing. In contrast with traditional neural networks, deep learning requires large volumes of data, regarding both features and samples, for proper training (Goodfellow, Bengio et al. 2016). Given the recent progress in activation functions, optimisation algorithms and hyperparameters, deep learning networks have outperformed shallow neural networks. Below is a summarised explanation of deep learning and its most important terminologies.

2.3.2. Model Representation Mathematics

Figure 2.5 represents a scheme of a simple Neural Network model with the following features: a series of input nodes (three input nodes and one bias), a hidden layer comprising four nodes and a single output node. Bias is a constant value that allows us to shift the output of the node to either left or right, similar to the role of a constant in a linear function. This adjusts the output a little to predict the data more accurately.

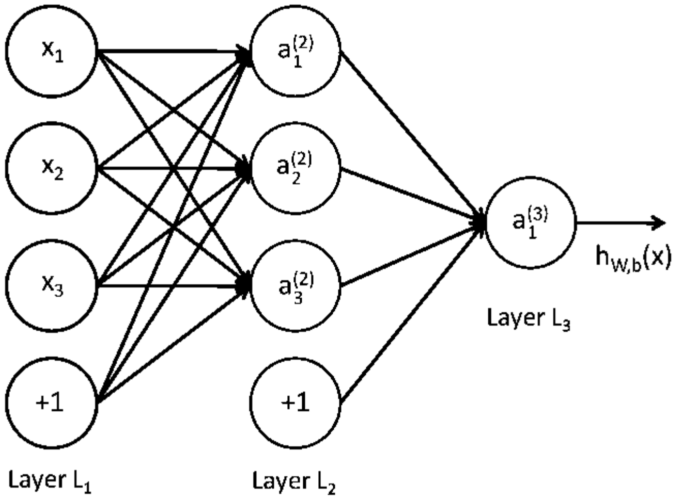


Figure 2.5, Schematic representation of a simple Neural Network model

By denoting the "bias" nodes in layers L_2 & L_1 as x_0 and a_0 respectively, we can insert the input nodes into a vector "X" and the hidden layer into a vector "A". It should be noted that x_0 and a_0 refer to the respective biases.

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}, \qquad A = \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

The weights (arrows) of the neural network are denoted by "w". As can be seen in the neural network diagram above, the weights between the input and hidden layer will be of a 3×4 matrix, with the weight between the hidden and output later being a 1×4 matrix.

$$\mathrm{w}^{(1)} = \begin{bmatrix} \mathrm{w}_{1,0} & \mathrm{w}_{1,1} & \mathrm{w}_{1,2} & \mathrm{w}_{1,3} \\ \mathrm{w}_{2,0} & \mathrm{w}_{2,1} & \mathrm{w}_{2,2} & \mathrm{w}_{2,3} \\ \mathrm{w}_{3,0} & \mathrm{w}_{3,1} & \mathrm{w}_{3,2} & \mathrm{w}_{3,3} \end{bmatrix}$$

The output for each hidden layer is calculated as follows: the input vector **X** is multiplied by the weights matrix $\boldsymbol{\theta}^{(1)}$**,** then the activation function **"g"** is applied to their product:

$$a_1^{(2)} = g\left(\mathrm{w}_{1,0}^{(1)}\, x_0 + \mathrm{w}_{1,1}^{(1)}\, x_1 + \mathrm{w}_{1,2}^{(1)}\, x_2 + \mathrm{w}_{1,3}^{(1)}\, x_3\right)$$

$$a_2^{(2)} = g\left(\mathrm{w}_{2,0}^{(1)}\, x_0 + \mathrm{w}_{2,1}^{(1)}\, x_1 + \mathrm{w}_{2,2}^{(1)}\, x_2 + \mathrm{w}_{2,3}^{(1)} x\ _3\right)$$

$$a_3^{(2)} = g\left(\mathrm{w}_{3,0}^{(1)}\, x_0 + \mathrm{w}_{3,1}^{(1)}\, x_1 + \mathrm{w}_{3,2}^{(1)}\, x_2 + \mathrm{w}_{3,3}^{(1)}\, x_3\right)$$

An activation function maps the resulting linear values into usually non-linear space. (This will be explained further in Section 2.3.3).

When a neural network has multiple hidden layers and nodes, we would obtain the following equation:

$$a_n^L = \left[\sigma\left(\sum\nolimits_m \mathrm{w}_{nm}^2 \left[\cdots\left[\sigma\left(\sum\nolimits_j \mathrm{w}_{kj}^2 \left[\sigma\left(\sum\nolimits_i \mathrm{w}_{ji}^1\, x_i + b_j^1\right)\right] + b_k^2\right)\right]\ldots\right]_m + b_n^L\right)\right]_n$$

where Layer **L** and layer **L-1** have **n** and **m** nodes, respectively.

If the right number of hidden layers, neurons per layer and a suitable activation function are selected, it allows the correlations of any significant data to be observed. In relation to improving the neural network design, there are a variety of parameters that can be changed and implemented. These will be discussed below.

2.3.3. Activation Functions

The activation function in neural networking refers to whether or not a given node should be triggered. As explained, this is decided based on the weighted sum of a node**.** Some of the more popular activation functions will be discussed below:

- Linear Functions

As the name suggests, using a linear function in neural networks would result in the output later being a linear function. This, however, means that it is not possible to map any non-linear data, as can be observed in Figure 2.6.

$$\boldsymbol{y = ax}$$

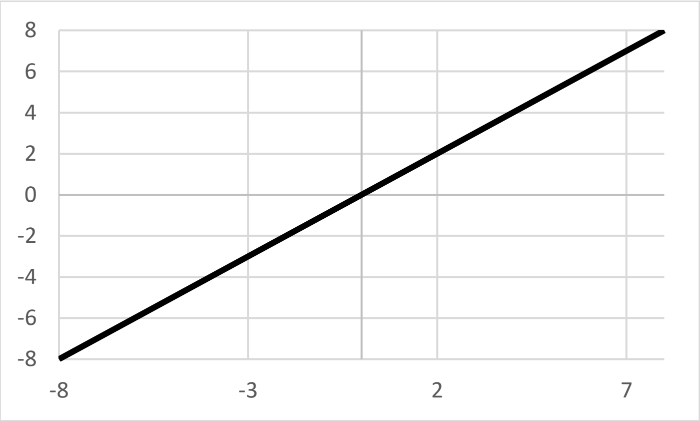


Figure 2.6, Linear activation function

- Sigmoid Function:

Sigmoid functions, better known as logarithmic functions, are a very well-known and popular approach as an activation function in neural networks, Figure 2.7. In this approach, the respective input of the function is transformed into a value between the range of 0.0 – 1.0. When an input value is significantly larger than 1.0, it is converted back to 1.0; likewise, for values considerably smaller than 0.0 which are rounded up to 0.0. For all possible inputs, the curve of this function is permeably a lazy S.

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

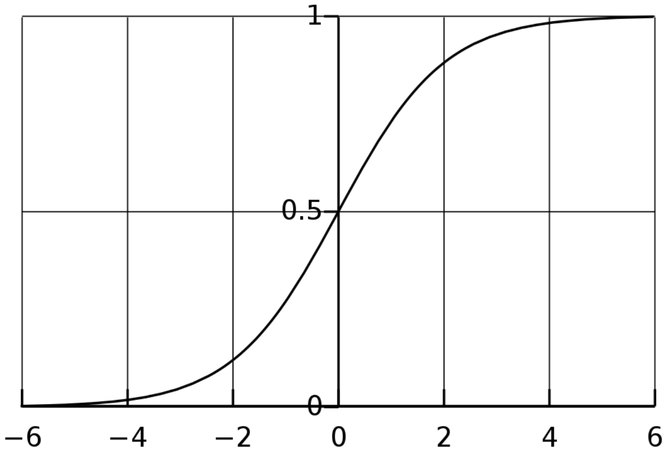


Figure 2.7, Sigmoid activation function

This function has limited sensitivity and saturation regardless of the information provided to the node. Once the information becomes saturated, it becomes very challenging for the algorithm to attempt to improve the performance of the model continuously via weight manipulation.

- ReLU

A Rectified Linear Unit (ReLU) is typically the most common activation function that is used throughout neural networks. Mathematically, the ReLU function is defined as $y = \max(0, x)$; graphically it can be observed in Figure 2.8.

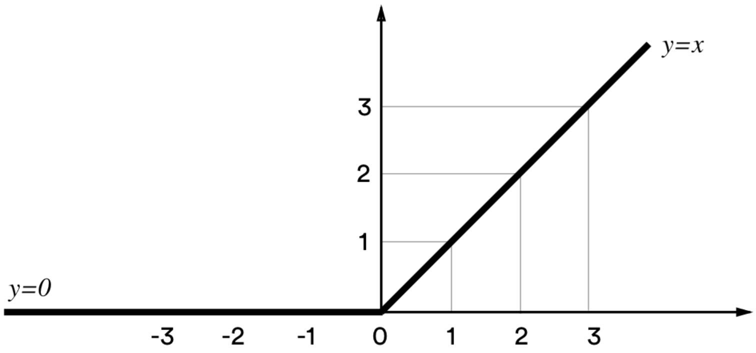


Figure 2.8, ReLU activation function

If the best activation function to use is unknown, the ReLU function is typically a sufficient first choice.

- ELU

The Exponential Linear Unit (ELU), is very similar to ReLU except for having negative inputs, as described below:

$$R(z) = \begin{Bmatrix} z & z > 0 \\ \boldsymbol{\alpha} \ (\ \mathrm{e}^{\mathrm{z}} - 1) & z \leq 0 \end{Bmatrix}$$

In terms of the cost function (this will be explained in Section 2.3.5), ELU tends to converge to zero more quickly and provide more accurate results than other functions. Compared with other activation functions, the ELU has an extra alpha constant, and this value should be a positive number.

In comparison with ReLU's sharp linearity at zero, ELU curves tend to smooth slowly until the output equals – α, as per Figure 2.9.

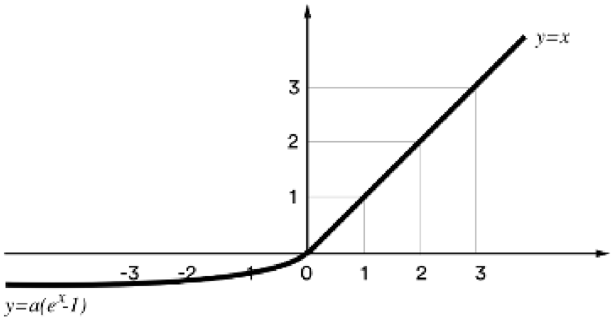


Figure 2.9, ELU activation function

2.3.4. Model Hyperparameters

Model hyperparameter is the term used to describe properties that govern the entire training process. These hyperparameters are made up of variables that are used to both determine the structure of the network (i.e., No. hidden units) and how the network is to be trained (i.e, the. learning rate). They are defined before the training, i.e., before the optimisation of weights and biases. Therefore, they are directly related to the performance given by a given model under training. Therefore, the appropriate hyperparameters must be chosen carefully, given their impact on the network. For example, given a low learning rate, the model may miss the important patterns presented in the data; whereas, if it is too high, it may result in collisions. The two main benefits of choosing good parameters are: first, that they allow the model to search efficiently across the space of possible hyperparameters, and second, that they enable easier management of a large set of experiments during parameter tuning. The most important hyperparameters are as follows:

- Learning Rate

The learning rate of a model refers to the size of each step taken per iteration. With a high learning rate, the large step size allows for more ground to be covered per iteration. However, this comes with the risk of overshooting the local minima, and can result in the algorithm potentially not converging. On the contrary, if a lower learning rate is used, we can confidently move towards the local minima. This is due to the sheer number of gradients that are calculated via the small step size. Low learning rates, although being more precise, have the downside of taking a significantly longer time in computing the minima of the gradient. Figure 2.10 illustrates the effect of different learning rates on the validation error. A common starting learning rate = 0.001.

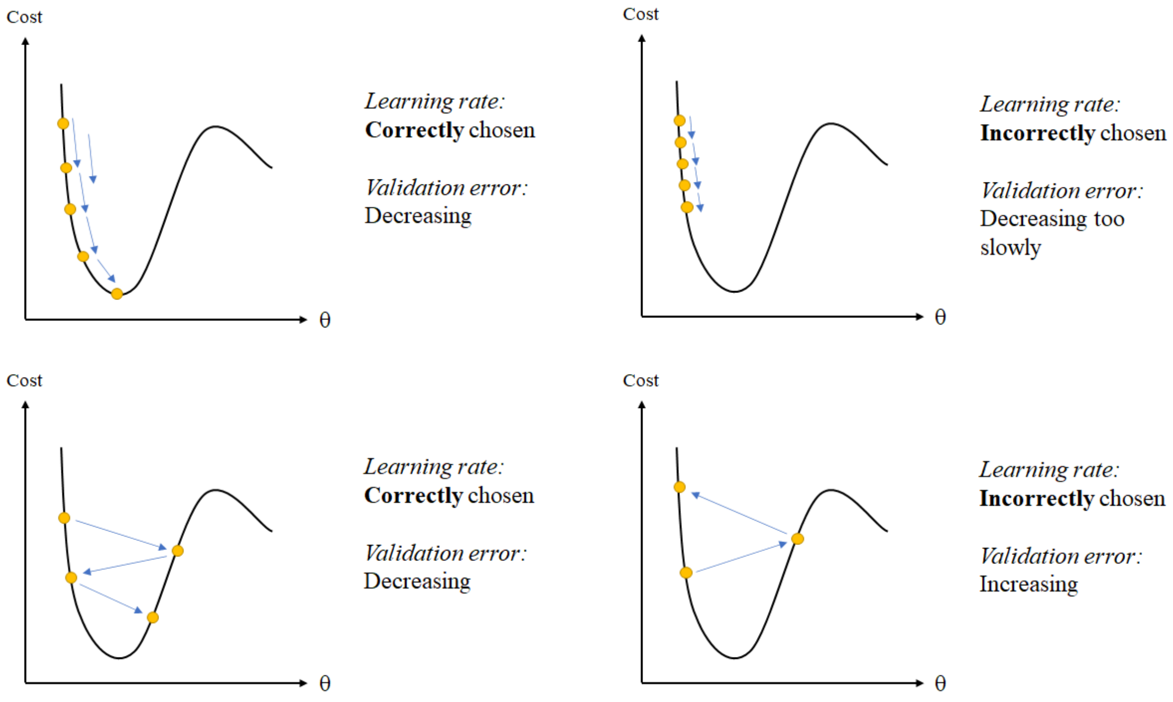


Figure 2.10, Illustrations of the effects of suitable and unsuitable learning rates on the validation error

- Mini-Batch Size:

Batch sizes have an influence on the following properties: the number of required resources in the training procedure, and the speed and number of iterations in a non-trivial system. The most common technique used today is the setting of a mini-batch size. If the mini-batch size is set to a value of 1, this is called stochastic training; on the other hand, if the mini-batch size is set to the number of examples in the training set, it is called batch training. The following starting values are recommended for use in experiments: 1, 2, 4, 8, 16, 32, 64, 128 and 258.

A larger mini batch causes a boost in computation; this boost, in turn, allows for the exploitation of matrix multiplications in training calculations. However, this comes with its consequences, requiring more memory in the training process. On the other hand, typically, a smaller mini batch introduces a higher amount of noise within error calculations; however, this can be used to prevent the training procedure from stopping at a local minimum. Hence, a typical safe value for minimal batch size = 32.

- Number of hidden units (model capacity)

The number of hidden units present within the neural network, determines the degree of the model's learning capacity. The learning capacity of a model refers to its ability to learn the respective function to approximate better. This demonstrates that, although the number of hidden units is a subtle hyperparameter, it should not be overlooked. Models with higher complexities typically have a larger number of hidden units and can fit a wide range of data; this is due to the learning capacity that this model requires.

When compared with the optimal number of units, having slightly more is not typically an issue; however, a significantly larger amount leads to overfitting the data, thereby reducing its ability to generalise.

- Number of epochs and generalization

Given the importance of overfitting and number of training iteration, the model performance must be monitored for both the training set (the data used to adjust the weights) and the validation set (the unseen data used to evaluate the model generalization), Figure 2.11. This figure demonstrates the relationship between the number of epochs and the concept of generalization, by plotting the errors associated with the training and validation sets as a function of the number of epochs. At lower epochs, errors for both the training and validation sets are high. However, after a particular point in the plot at higher epochs, while the training error continuously decreases, the validation error starts to increase as a result of overfitting.

Therefore, that particular turning point is the optimal number of epochs for this model, which represents the optimal generalization for this training set.

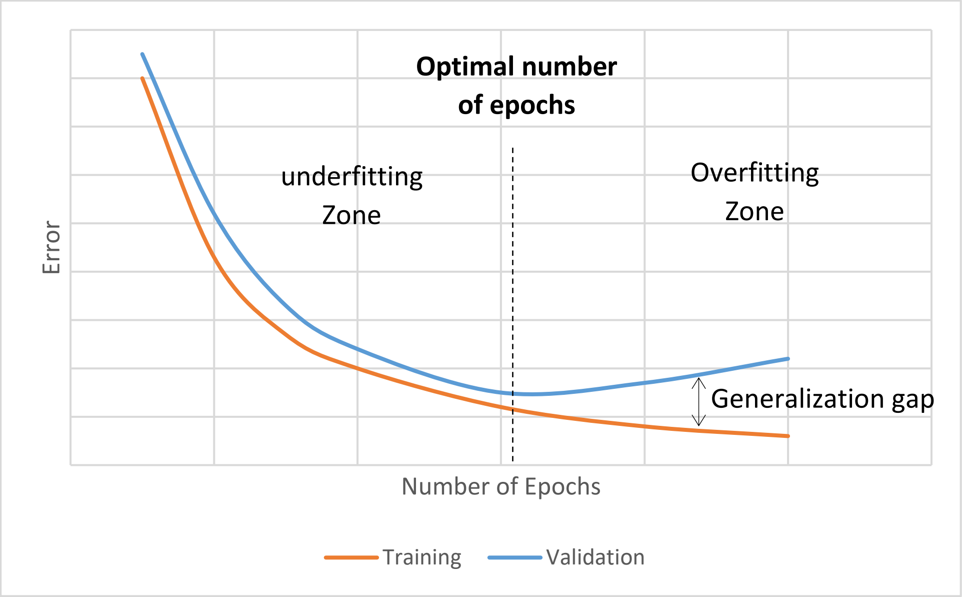


Figure 2.11, Illustration of the effect of the number of epochs on the validation error

A common method in monitoring the validation is through intuitive methods. This method involves the close monitoring of the validation error and stopping the model only once the validation error stops decreasing. This technique, called "early stopping", involves stopping the training process if, within the past 10-20 epochs, the validation error has not been improved.

2.3.5. Optimiser Algorithms

- Gradient Descent

Gradient descent refers to an optimisation algorithm that allows the minimization of a cost function and finds the optimal parameters associated with this minimised function. This is achieved by iteratively taking steps in the direction of steepest descent, as is defined by the negative gradient, Figure 2.12. Gradient descent is used in machine learning to update the parameters of the respective model. In this case, the parameters refer to both the coefficients in linear regression as well as the weights and biases of the neural networks.

Gradient descent begins at an initial value, corresponding with random values of weights and biases, and starts to move in the direction of the steepest descent. Due to the shape of the cost function (convex), initialisation does not affect the outcome. Each step tries to reduce the value of the cost and takes values towards the downward direction, as much as possible. After a series of iterations, gradient descent will converge towards a minimum cost, that associated with optimum network parameters. The learning rate, as described earlier, controls the size of step taken in each direction.

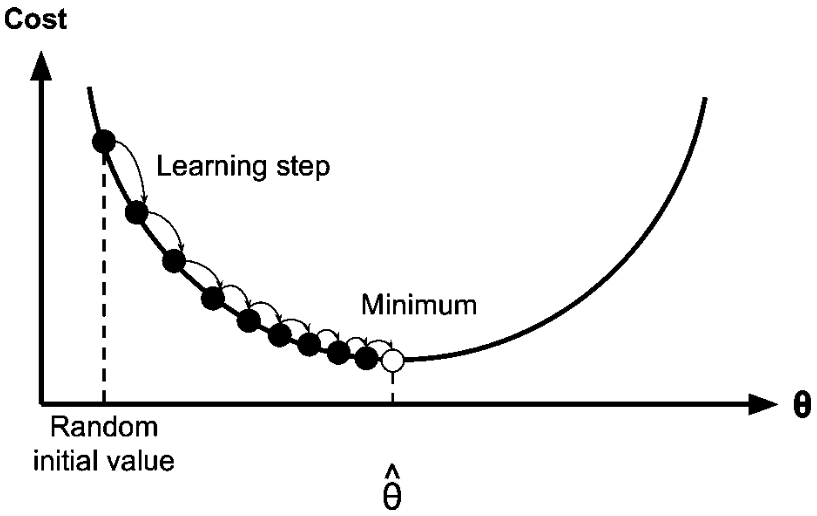


Figure 2.12, Illustration of the gradient descent algorithm

- Backpropagation

Backpropagation is the name given to an algorithm that is used in the training of artificial neural networks. This is achieved through the analysis of gradient descent. Once the artificial neural network and its respective errors are provided, this method is able to calculate the gradient present in the error function with respect to the overall neural network weight. This has been observed to be a generalisation of the delta rule, where perceptions are used to update parameters in the multilayered neural network.

$$\theta^{t+1} = \theta^{t} - \eta \frac{\partial E(X, \theta^{t})}{\partial \theta}$$

Where:

- E(X, θ^{t}) is the gradient of the error function, with respect to the weight $w_{i,j}^{k}$ and biases b_{i}^{k}
- $\eta = the\ learning\ rate$
- $\theta = the\ collective\ of\ weights\ w_{i,j}^{k}\ and\ biases\ b_{i}^{k}$
- $\theta^{t} = parameters\ of\ the\ neural\ network\ at\ iteration\ 't'$

This 'backwards' aspect of the 'backpropagation' method, refers to the backward direction that the network follows to calculate the gradient. This implies that the gradient in the last layer is computed first, and consequently, the gradient found in the first layer is determined last. Throughout this method, the partial calculations of the gradient from each layer, are employed in the calculation of the next layer (in the backwards direction). This back stream of error data provides the means for effective computation of the respective gradient per layer compared with the separate calculation of each respective gradient layer.

- Cost function

A loss function allows an accurate analysis of a model’s prediction for a single training set. A cost function defines as the average of the loss function over the entire training set and has its respective curve and gradients. Based on the slope of this curve, parameters should be updated to increase the accuracy of the model.

Given the cost function:

$$f(\mathrm{w}, b) = \frac{1}{N} \sum_{i=n}^{n} \left(y_i - (\mathrm{w}x_i + b)\right)^2$$

The gradient can be calculated as:

$$f'(\mathrm{w}, b) = \begin{bmatrix} \frac{df}{d\mathrm{w}} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{i=n}^{n} -2x_i\big(y_i - (\mathrm{w}x_i + b)\big) \\ \frac{1}{N}\sum_{i=n}^{n} -2\big(y_i - (\mathrm{w}x_i + b)\big) \end{bmatrix}$$

As described, by running the gradient descent using the cost function, two parameters can be controlled. These are the weights "w" and the bias "b". As each of these has a significant impact on the final prediction, partial derivatives can then be calculated, each with respect to the parameters, to be analysed as gradients.

To solve for the gradient of each parameter, the data points must be iterated using "w" and "b" values in order to calculate the partial derivatives. This new gradient can then show the slope of the cost function at the specified position alongside the direction that the updated parameters should head towards. The learning rate controls the size of each update.

- Stochastic Gradient Descent

The stochastic gradient descent (SGD) is a fundamental algorithm used for converging neural networks. There are several gradient descent algorithms. One of the main differences among these algorithms is the number of cost calculations taken per step. As only one example is needed to calculate SGD, using this algorithm significantly increases the speed of the neural networks.

Below is the SGD equation: it enables updating of parameters within a neural network. The parameters are updated in a backwards pass via backpropagation to calculate the gradient ∇:

$$\theta = \theta - \eta . \nabla_{\theta} \, \mathrm{J}(\theta; \mathrm{x}, \mathrm{y})$$

where:

- J is the known objective function (cost function/loss function)
- θ are the parameters
- η is the learning rate
- $\nabla_{\theta} J(\theta; x, y)$ is the backpropagation value

$(\theta; x, y)$ means that the parameters (θ) have to be inputted alongside a training example and a label, denoted 'x' and 'y' respectively. The semicolon in this equation indicates that the parameter theta θ is different from the training example (x) and the label (y), which are divided by a comma.

The advantage of using SGD is that its process is relatively quick compared with the other gradient descent approaches. SDG is also relatively easy to learn due to its non-mathematics intensive approach when compared with newer models.

However, as SDG is an old relative algorithm in optimisation, it is typically slower to converge than a newer algorithm. SGD also has a higher likelihood of being trapped at a local minimum rather than at the actual minimum. It is also outperformed in terms of optimising the cost function by more recent approaches.

- Momentum gradient descent

The momentum algorithm can be used to obtain to a local minimum faster, which can increase the speed of progression. To do this, a temperate element of time is inserted in the equation used to update the parameter of a neural network.

This time element increases the momentum of the parameter's change by some gamma amount 'γ', which is typically initialized to 0.9. This gamma value is then multiplied by the previous update ' ν_t'.

$$\theta = \theta - \eta\nabla J(\theta) + \gamma\, \nu_t$$

It can be observed that the momentum equation is very similar to the SGD equation, but with an added term.

Where:

- J is the known objective function (cost function/loss function)
- θ are the parameters
- η is the learning rate
- γ is the momentum term
- ν_t is the last update to θ, called the previous update.

- Adaptive moment estimation (Adam)

The adaptive moment estimation, also known as Adam, is most typically the optimiser, which performs the best. This is done using momentum and adaptive learning rates so as to converge faster.

This optimisation algorithm keeps both averages of past gradient m and past squared gradient v at decay rates of β. Given the parameters of w^t and the loss functions of L^t, where 't' denotes the index of the current training iteration (beginning at 0), Adam's parameter updates are given by the following:

$$m_\theta^{t+1} = \beta_1 m_\theta^t + (1-\beta)\nabla_\theta L^t$$

$$v_\theta^{t+1} = \beta_2 v_\theta^t + (1-\beta_2)(\nabla_\theta L^t)^2$$

m_w^{t+1} and v_w^{t+1} are estimations of the first and second moments, or, in other words, the mean and the uncentered variance of the gradients, respectively. During initial time iterations, when the decay rates are low, these values are biased towards zero. To lower these biases, the following corrections are applied:

$$\widehat{m}_\theta = \frac{m_\theta^{t+1}}{1-\beta_1^{t+1}}$$

$$\hat{v} = \frac{v_\theta^{t+1}}{1-\beta_2^{t+1}}$$

Using these corrected values, the desired parameter (weight or bias) can be calculated for the next time step:

$$\theta^{t+1} = \theta^t - \eta\left(\frac{\widehat{m}_\theta}{\sqrt{\hat{v}_\theta} + \epsilon}\right)$$

where ϵ is a smoothing term to prevent division by zero.

- Adaptive gradients (AdaGrad) algorithm

Adaptive Gradients (AdaGrad) is a relatively simple approach where the learning rate is changed over time. This allows adaptation to the differences in datasets, considering there can be small or large updates accordingly to the definition of the learning rate. Having an adaptive learning rate enables large steps to be taken initially, followed by smaller steps as the local minimum is approached. This allows for an initially fast move, followed by smaller and smaller steps as the learning rate decays, allowing for the faster convergence, and thus the local minimum will not be overstepped.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{\varepsilon + \sum_{\tau=1}^{t}\left(\nabla J\left(\theta_{\tau,i}\right)\right)^2}} \nabla J\left(\theta_{t,i}\right)$$

Where:

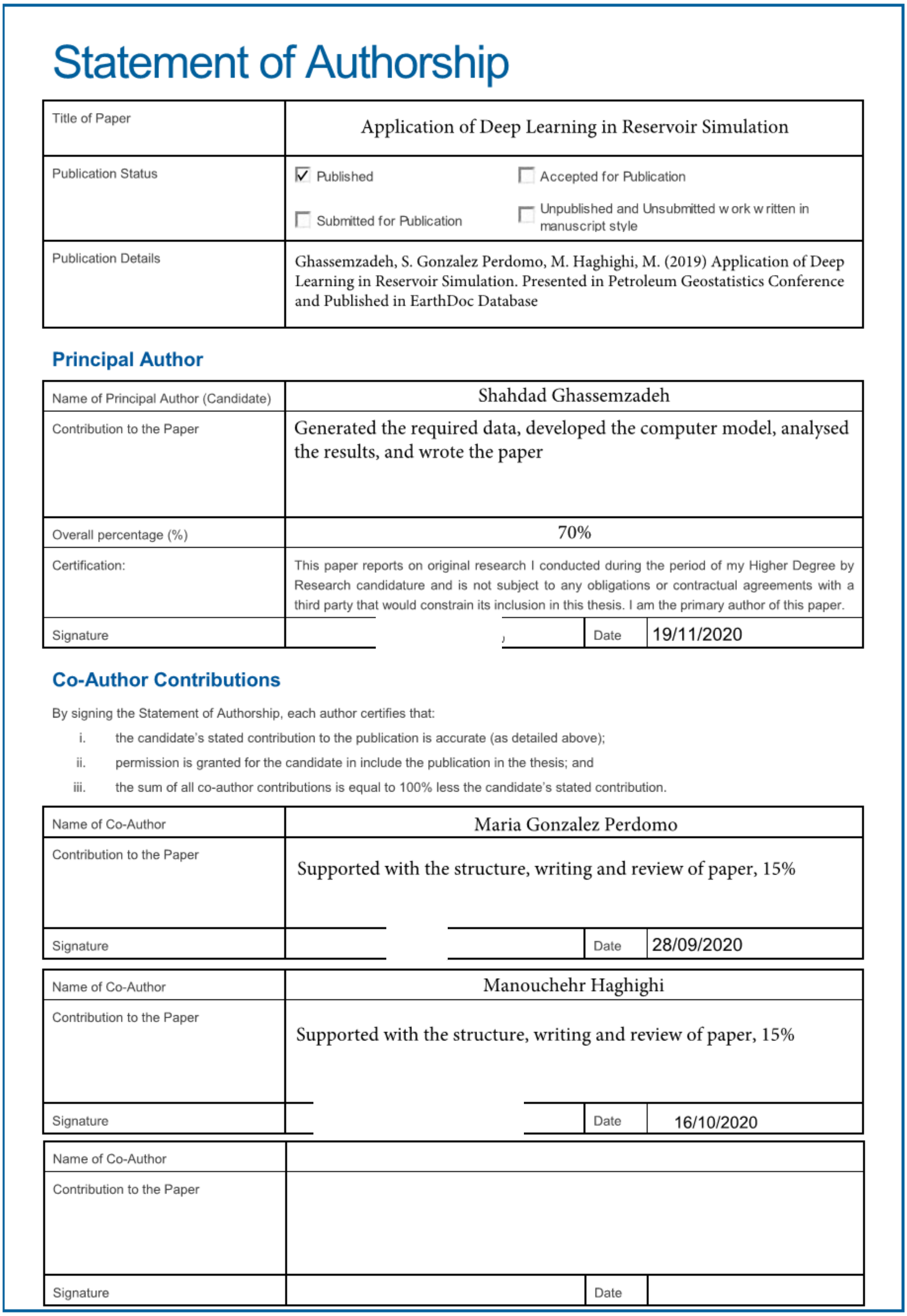
- ε is a small value to ensure that it does not divide by zero

- $\sqrt{\sum_{\tau=1}^{t}\left(\nabla J\left(\theta_{\tau,i}\right)\right)^2}$ is the square root of the summation over the gradients squared. These gradients are summed from time step $\tau = 1$ up to current time step 't'.

This algorithm works best for sparse data, as the learning rate is decreased more quickly for frequent parameters, and more slowly for infrequent parameters. However, there are cases where the learning rate can decrease very fast due to the accumulation of gradients from the training, showing that there is a point where the model will not learn again due to the learning rate being almost zero.

3. A feasibility study into the application of Deep Learning in Reservoir Simulation

3.1.Application of deep learning in reservoir simulation



Statement of Authorship

Title of Paper	Application of Deep Learning in Reservoir Simulation		
Publication Status	☑ Published □ Submitted for Publication	□ Accepted for Publication □ Unpublished and Unsubmitted w ork w ritten in manuscript style	
Publication Details	Ghassemzadeh, S. Gonzalez Perdomo, M. Haghighi, M. (2019) Application of Deep Learning in Reservoir Simulation. Presented in Petroleum Geostatistics Conference and Published in EarthDoc Database		

Principal Author

Name of Principal Author (Candidate)	Shahdad Ghassemzadeh		
Contribution to the Paper	Generated the required data, developed the computer model, analysed the results, and wrote the paper		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	19/11/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Maria Gonzalez Perdomo		
Contribution to the Paper	Supported with the structure, writing and review of paper, 15%		
Signature		Date	28/09/2020

Name of Co-Author	Manouchehr Haghighi		
Contribution to the Paper	Supported with the structure, writing and review of paper, 15%		
Signature		Date	16/10/2020

Name of Co-Author			
Contribution to the Paper			
Signature		Date	

Please cut and paste additional co-author panels here as required.

WeP07

Application of Deep Learning in Reservoir Simulation

S. Ghassemzadeh[1]*, M. Gonzalez Perdomo[1], M. Haghighi[1]

[1] University of Adelaide

Summary

Reservoir simulation plays a vital role as oil and gas companies rely on them in the development of new fields. Therefore, a reliable and fast reservoir simulation is a crucial instrument to explore more scenarios and optimize the production. In each simulation, the reservoir is divided into millions of cells, and rock and fluid attributes are assigned to these cells. Then, based on these attributes, flow equations are solved with time-consuming numerical methods. Given the recent progress in machine learning, the possibility of using deep learning in reservoir simulation has been investigated in this paper. In the new approach, fluid flow equations are solved using a deep learning-based simulator instead of time-consuming mathematical approaches. In this paper, we studied 1D Oil Reservoir and 2D Gas Reservoir. Data sets generated using the numerical models were used to create the developed simulators. We used two metrics to evaluate our models: Mean Absolute Percentage Error (MAPE) and correlation coefficient (R2). Given the low value of these matrics (MAPE < 15.1%, R2 >0.84 for 1D and MAPE < 0.84%, R2 ≈ 1 for 2D), the results confirmed that the deep learning approach is reasonably accurate and trustworthy when compared with mathematically derived models.

Introduction

In reservoir engineering, the main aim of modeling is to provide an acceptable understanding of the future performance of fluid flow by estimating parameters such as oil recovery under a specific production strategy. The integration of the simulation output with the operating expenses and revenues forecast will assist engineers with the generation of the future cash flow predictions.

Although introducing powerful hardware and software in recent years have improved the reservoir models and simulation undoubtedly, computational inefficiency and slow running time are still vital problems in different reservoir simulation areas such as history matching and optimization applications. Besides, model dimensions are surpassing a billion cells. Considering the number of scenarios that need to be examined to find the optimized parameters, the reservoir optimization, and management processes are getting more computational-expensive every year. Therefore, one of the main priorities in the oil industry is having a faster reservoir simulation.

Reduced-order modeling (ROM) is a popular method that addresses the previously mentioned concerns. The main idea behind this approach is to find an input-output relationship, which is approximately equal to the input-output description of the fine model by projecting the large-scale states onto a much smaller subspace (Ghasemi, Ibrahim, & Gildin, 2014; Heijn, Markovinovic, & Jansen, 2004).

To overcome the time-step problem and reduce the dimensionality of the discrete problem, Sheth and Younis proposed an adaptive solution strategy for the multicomponent simulation of general fluid flow problems (Sheth & Younis, 2017). Moreover, tuning and optimizing the simulator can speed up the simulation performance up to six times (Eldred et al., 2014; Tleukhabyluly, Dallorto, Porcelli, & Tarantini, 2016). Besides, running the model on a multi-core computer (Supercomputer) can even provide a faster approach, although there is a considerably associated cost (Beckner, Haugen, Maliassov, Dyadechko, & Wiegand, 2015; Eldred et al., 2014; Guan et al., 2015).

Another approach to this is machine-learning techniques. These techniques are widely used in the petroleum industry in different research areas such as Formation Evaluation (Sbiga & Potter, 2017; Siregar, Niu, Mostaghimi, & Armstrong, 2017), Prediction of Physical Properties of Hydrocarbons (Ghassemzadeh, Schaffie, Sarrafi, & Ranjbar, 2014; Ghassemzadeh, Shafflie, Sarrafi, & Ranjbar, 2013) , Well placement (Bangerth, Klie, Wheeler, Stoffa, & Sen, 2006; Handels, Zandvliet, Brouwer, & Jansen, 2007) and Production Optimization (Ghassemzadeh & Charkhi, 2016; Ghassemzadeh & Pourafshary, 2015).

However, all of these approaches have some significant downsides and cannot fully satisfy the need of a fast and reliable simulator. As Deep Learning has shown significant improvement in previously failed models, we investigated the possibility of using this method to improve the efficiency of reservoir simulators. The novelty of this work lies in the use of deep learning to develop a new type of artificial intelligence based simulator. An advantage of using such simulator is that it enables us to investigate more scenarios and present a more profitable production scheme.

Method

The main aim of reservoir simulation is to estimate and forecast different recovery scheme to evaluate the effects of altered operating conditions and compare them economically. Clearly, in order to explore more scenarios, it is necessary to reduce the running time of the simulation.

In the proposed approach, the main idea is similar to what a commercial reservoir simulator does. However, to reduce the run time significantly, we are eliminating the most time-consuming task in the simulation process. Thus, the main aim is to replace the procedure of solving fluid flow equations with a deep-learned based network.

In real case scenario, we mostly have to deal with a compositional multiphase flow in a three-dimensional space, which makes the problem very complicated to model. Therefore, to investigate the practicality of the idea and understand the effect of each variable, we started from the simplest case, a 1D single-phase reservoir. Simulation for 1D reservoir has a limited application in the oil industry, therefore, in the more realistic scenario, we developed a simulator for 2D gas reservoirs.

In summary, any reservoir simulator models the production of multiple wells in an oil field as a function of different variables such as reservoir and fluid characteristics, well and surface facilities configuration, geology conditions and operational constraints, which can be formulated simply as:

$$q = f(X1, X2, \ldots, Xn) \qquad \text{Equation 1}$$

Where q is the production rate, and Xs are different variables such as well data, fluid and rock properties and production data.

Since we are dealing with a data-driven model, we need a source of a reliable dataset. Thus, our model is made of two parts:

- Data generating process, to produce data using a mathematical model or commercial simulator.
- The data-driven simulator, to create an intelligent network, which is capable of learning and understanding the fluid flow phenomenon, using a deep learning model which would be validated it under different scenarios.

In any data-driven model, the quality of the input data is a critical parameter in the accuracy of the final model. This need to be highlighted, since it has been the first real attempt to create a standalone data-driven reservoir simulator. Therefore, we kept the model as simple as possible to understand the problem better. Besides, to avoid neglecting any factors that affect fluid flow and consider all the variables that relate the output to the inputs, we did not use the commercial simulator in the first step. Therefore, we created a 1D oil reservoir model. Considering a single production well in our system, we solved the flow equation using the fully-implicitly numerical method. The developed model can predict the pressure of each grid in a 1D single-phase reservoir through different time-steps.

Then, to study the application of deep learning in a more sophisticated scenario, we used Schlumberger Eclipse, Black Oil Simulator, to create a 2D gas reservoir model and generate the required dataset. This software uses a more complicated but realistic fluid flow equation. Parameters such as compressibility of rock and fluid and residual saturation are considered in its built-in fluid equation. Similar to the first case, a fully implicit numerical method was used to avoid any convergence problem. It was assumed that there was only one production well in the field.

The next step was to create a deep learning simulator for a 2D reservoir. Therefore, it was necessary to divide the dataset into a learning, validation, and test data sets. The most critical section in this research was the network architecture. In order to find the best network, we started from two dense layers and studied nets up to 25 layers. For each net, we optimized the network using different hyper parameters and activation functions.

Results and discussion

In this research, a 1D and 2D reservoir simulators were developed, the first one saturated with oil and the second one saturated with gas and water. In both cases, the dynamic fluid is single phase, and it is produced from a single well. The main idea, is to create a data-driven simulator, capable of learning the physics of the fluid flow and acting like a stand-alone simulator, not just a simple interpolator among the dataset. To do so, a dataset was generated using numerical models, and then the data was divided into the training, validation and test data sets.

Using the generated data for 1D, we created a deep-learning based network, which was optimized, starting from two layers up to 25 layers. Based on results of different nets, the network with fifteen layers provide the best results, while increasing the number of layers to more than 15 do not provide any significant advantages.

Table 1, comparing the accuracy of the different sets for different Scenario

	1D			2D		
	Learning data set	**Validation data set**	**Test data set**	**Learning data set**	**Validation data set**	**Test data set**
MAPE	14.3	15.1	14.9	0.82	0.82	0.84
MAE	28.9	29.6	29.7	19.07	19.17	19.1
R2	85	84.1	84.3	1	1	1

The comparison of the accuracy among learning, validation and test datasets are shown in table 1. The low value of MAPE of the results for 1D (14.3 – 15.1) indicates that the developed simulator can extract the existed pattern among the data points and obtain adequate understanding between the inputs and desired output.

Results of the developed model for the 1D oil reservoir, show that the idea of using deep learning to develop a simulator is worthy enough to investigate more. To do so, we used the generated data and taught the physics of 2D fluid flow to a 9-layer network. In this scenario, the results were even more accurate. MAPE for training, validating and test data are 0.82, 0.82 and 0.84 respectively, MAE for these sets are 19.07, 19.17 and 19.10 Psi respectively, and R2 in all sets are almost 1.

Conclusions

The long running time of reservoir simulation has been an ongoing concern for the oil industry, since it makes the history matching and production optimization an incredibly time-consuming and challenging task. Several solutions have been proposed to make each simulation more cost-effective, and speed up the runtime of simulation.

In the current study, we explored the possibility of replacing the fluid flow equations with a data-driven network. Based on this idea, we developed an optimized stand-alone reservoir simulator for 1D and 2D single-phase flow in the porous media. Developed models estimate the pressure drop throughout the reservoir using the same inputs as a mathematical based simulator.

A stand-alone deep learning-based simulator for modeling hydrocarbon flow in a sub-surface media has not been previously presented. The significant contributions of our current study are summarized as follows:

1. This study addressed the possibility of replacing a complex mathematical reservoir model with a deep neural network, to improve computational efficiency; increase accuracy, and mainly decrease the simulation running time.
2. The study used deep learning and developed a stand-alone reservoir simulator, by which the pressure drop distribution throughout the reservoir could be estimated with an acceptable margin of error.
3. The proposed model presented a novel approach for modeling fluid flow in a porous media based on fluid and rock properties and production characteristics.
4. The study presented a fast approach leading to explore more scenarios in history matching and optimization. Having explore more scenarios, engineers can provide better future cash flow prediction and further benefit the company.

The proposed simulators were examined with various examples, and in all cases, it was able to understand the fluid behaviour in each grid accurately and predicts the pressure with acceptable accuracy (MAPE< %4 for 1D and MAPE<2.3 for 2D).

Acknowledgments

The authors are grateful to Santos Ltd for their financial support in this research.

References

Bangerth, W., Klie, H., Wheeler, M. F., Stoffa, P. L., & Sen, M. K. (2006). On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences, 10*(3), 303-319. doi:10.1007/s10596-006-9025-7

Beckner, B. L., Haugen, K. B., Maliassov, S., Dyadechko, V., & Wiegand, K. D. (2015). *General Parallel Reservoir Simulation*. Paper presented at the International Petroleum Technology Conference, Doha, Qatar.

Eldred, M. E., Orangi, A., Al-Emadi, A. A., Ahmad, A., O'Reilly, T. J., & Barghouti, N. (2014). *Reservoir Simulations in a High Performance Cloud Computing Environment*. Paper presented at the SPE Intelligent Energy Conference & Exhibition, Utrecht, The Netherlands.

Ghasemi, M., Ibrahim, A., & Gildin, E. (2014). *Reduced Order Modeling In Reservoir Simulation Using the Bilinear Approximation Techniques* (Vol. 2).

Ghassemzadeh, S., & Charkhi, A. H. (2016). Optimization of integrated production system using advanced proxy based models: A new approach. *Journal of Natural Gas Science and Engineering, 35*, 89-96. doi:https://doi.org/10.1016/j.jngse.2016.08.045

Ghassemzadeh, S., & Pourafshary, P. (2015). Development of an intelligent economic model to optimize the initiation time of gas lift operation. *Journal of Petroleum Exploration and Production Technology, 5*(3), 315-320. doi:10.1007/s13202-014-0140-z

Ghassemzadeh, S., Schaffie, M., Sarrafi, A., & Ranjbar, M. (2014). Predicting Dew Point Pressure: Using a Hybrid Intelligent Network. *Petroleum Science and Technology, 32*(24), 2969-2975. doi:10.1080/10916466.2014.919004

Ghassemzadeh, S., Shafflie, M., Sarrafi, A., & Ranjbar, M. (2013). The Importance of Normalization in Predicting Dew Point Pressure by ANFIS. *Petroleum Science and Technology, 31*(10), 1040-1047. doi:10.1080/10916466.2011.598895

Guan, W., Qiao, C., Zhang, H., Zhang, C.-S., Zhi, M., Zhu, Z., . . . Xu, J. (2015). *On Robust and Efficient Parallel Reservoir Simulation on Tianhe-2*. Paper presented at the SPE Reservoir Characterisation and Simulation Conference and Exhibition, Abu Dhabi, UAE.

Handels, M., Zandvliet, M., Brouwer, R., & Jansen, J. D. (2007). *Adjoint-Based Well-Placement Optimization Under Production Constraints*. Paper presented at the SPE Reservoir Simulation Symposium, Houston, Texas, U.S.A.

Heijn, T., Markovinovic, R., & Jansen, J.-D. (2004). *Generation of low-order reservoir models using system-theoretical concepts* (Vol. 9).

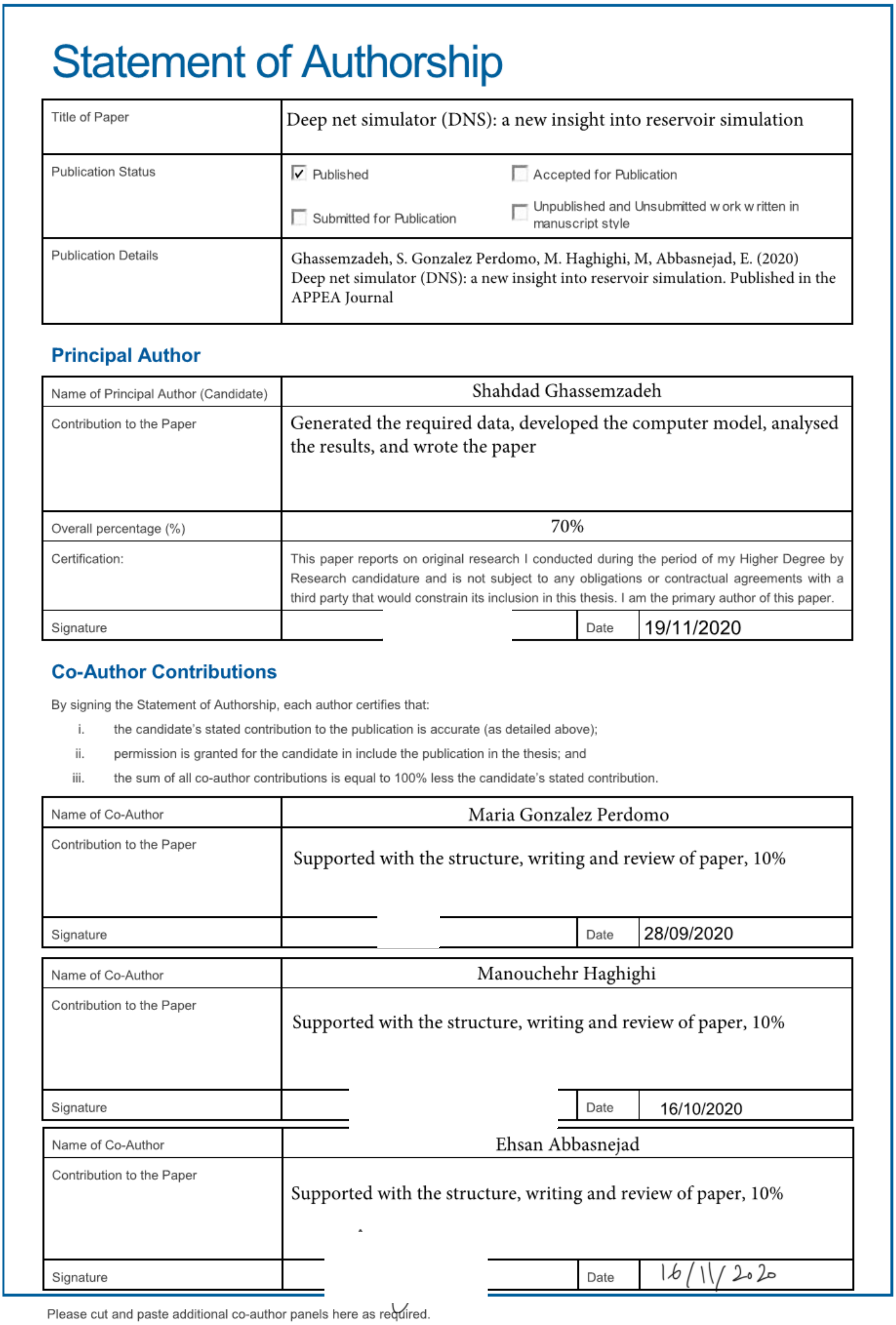
Sbiga, H. M., & Potter, D. K. (2017). Prediction of Resistivity Index by Use of Neural Networks With Different Combinations of Wireline Logs and Minimal Core Data. doi:10.2118/181751-PA

Sheth, S. M., & Younis, R. M. (2017). *Localized Solvers for General Full-Resolution Implicit Reservoir Simulation*. Paper presented at the SPE Reservoir Simulation Conference, Montgomery, Texas, USA.

Siregar, I., Niu, Y., Mostaghimi, P., & Armstrong, R. T. (2017). Coal ash content estimation using fuzzy curves and ensemble neural networks for well log analysis. *International Journal of Coal Geology, 181*, 11-22. doi:https://doi.org/10.1016/j.coal.2017.08.003

Tleukhabyluly, O., Dallorto, M., Porcelli, F., & Tarantini, V. (2016). *Speeding Up a Reservoir Simulation – Case Study on Giant Carbonate Reservoirs*. Paper presented at the SPE Annual Caspian Technical Conference & Exhibition, Astana, Kazakhstan.

3.2.Deep Net Simulator (DNS): a New Insight into Reservoir Simulation



Statement of Authorship

Title of Paper	Deep net simulator (DNS): a new insight into reservoir simulation
Publication Status	☑ Published □ Accepted for Publication □ Submitted for Publication □ Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	Ghassemzadeh, S. Gonzalez Perdomo, M. Haghighi, M, Abbasnejad, E. (2020) Deep net simulator (DNS): a new insight into reservoir simulation. Published in the APPEA Journal

Principal Author

Name of Principal Author (Candidate)	Shahdad Ghassemzadeh		
Contribution to the Paper	Generated the required data, developed the computer model, analysed the results, and wrote the paper		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	19/11/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Maria Gonzalez Perdomo		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	28/09/2020

Name of Co-Author	Manouchehr Haghighi		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	16/10/2020

Name of Co-Author	Ehsan Abbasnejad		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	16/11/2020

Please cut and paste additional co-author panels here as required.

The APPEA Journal 2020, **60**, 124–132
https://doi.org/10.1071/AJ19093

Peer reviewed paper

Deep net simulator (DNS): a new insight into reservoir simulation

Shahdad Ghassemzadeh[A,C], *Maria Gonzalez Perdomo*[A], *Manouchehr Haghighi*[A] *and Ehsan Abbasnejad*[B]

[A]Australian School of Petroleum and Energy Resources, Santos Petroleum Engineering Building, University of Adelaide, SA 5005, Australia.
[B]Australian Institute for Machine Learning, Engineering and Maths Sciences, North Terrace, University of Adelaide, Adelaide, SA 5000, Australia.
[C]Corresponding author. Email: shahdad.ghassemzadeh@adelaide.edu.au

Abstract. Reservoir simulation plays a vital role as a diagnostics tool to better understand and predict a reservoir's behaviour. The primary purpose of running a reservoir simulation is to replicate reservoir performance under different production conditions; therefore, the development of a reliable and fast dynamic reservoir model is a priority for the industry. In each simulation, the reservoir is divided into millions of cells, with fluid and rock attributes assigned to each cell. Based on these attributes, flow equations are solved through numerical methods, resulting in an excessively long processing time. Given the recent progress in machine learning methods, this study aimed to further investigate the possibility of using deep learning in reservoir simulations. Throughout this paper, we used deep learning to build a data-driven simulator for both 1D oil and 2D gas reservoirs. In this approach, instead of solving fluid flow equations directly, a data-driven model instantly predicts the reservoir pressure using the same input data of a numerical simulator. Datasets were generated using a physics-based simulator. It was found that for the training and validation sets, the mean absolute percentage error (MAPE) was less than 15.1% and the correlation coefficient, R^2, was more than 0.84 for the 1D oil reservoirs, while for the 2D gas reservoir MAPE < 0.84% and $R^2 \approx 1$. Furthermore, the sensitivity analysis results confirmed that the proposed approach has promising potential (MAPE < 5%, $R^2 > 0.9$). The results agreed that the deep learning based, data-driven model is reasonably accurate and trustworthy when compared with physics-derived models.

Keywords: deep learning, fluid flow, machine learning, proxy model, reservoir simulation, stand-alone simulator.

Received 6 January 2020, accepted 31 January 2020, published online 15 May 2020

Introduction

In reservoir engineering, the main aim of modelling is to provide an adequate understanding of the future performance of fluid flow by estimating parameters, such as oil recovery, under a specific production strategy. The integration of a subsurface simulation output with a multi-phase flow model in a wellbore and surface facilities is required for real-time production automation.

Although introducing powerful hardware and software in recent years has improved reservoir modelling and simulation, computational inefficiency and slow runtime are still ongoing problems in different reservoir simulation areas, such as history matching and optimisation applications. Furthermore, model sizes can now surpass a billion cells. Considering the number of scenarios that need to be examined to find the optimised parameters, reservoir optimisation and management processes are becoming more computationally expensive every year (Beckner *et al.* 2015). Therefore, one of the main priorities in the oil and gas industry is to create a faster reservoir simulator.

Reduced-order modelling (ROM) is a popular method that addresses the previously mentioned concerns. The main idea behind this approach is to find an input-output relationship that is approximately equal to the input-output description of the detailed model by projecting the large-scale states onto a much smaller subspace. ROM has been applied in reservoir simulation with different frameworks, such as the balanced truncation, proper orthogonal decomposition, trajectory-piecewise linear and dynamic mode decomposition (Cardoso and Durlofsky 2010; van Doren *et al.* 2006); however, it requires massive computational time, especially in more realistic scenarios, such as cases of complex geology (Li and Zhang 2007).

Another approach is machine learning, particularly artificial neural networks (ANN). These techniques are widely used in the petroleum industry in different research areas, such as formation evaluation (Sbiga and Potter 2017), reservoir thermodynamics (Ghassemzadeh *et al.* 2013, 2014; Shams *et al.* 2015), well placement (Bangerth *et al.* 2006) and production optimisation (Wood 2014; Ghassemzadeh and Pourafshary 2015). Furthermore, ANN has been used to create proxy models for different types of reservoirs, such as fracture (Artun *et al.* 2011), shale gas (Knudsen and Foss 2015), conventional (Goodwin 2015), enhanced oil recovery (Alenezi and Mohaghegh 2017) and field optimisation (Ghassemzadeh and Charkhi 2016; Mohaghegh *et al.* 2017).

The downside to these proxy models, however, is that each is built based on a corresponding specific reservoir model; this means that it can only be used as a replacement to this corresponding model. Since a proxy model acts only as a replacement to its corresponding reservoir, in the case of a new reservoir, the entire process of building a proxy must be restarted from scratch. This includes redoing the entire time-consuming numerical simulation for the new reservoir in order to build the new proxy model. It implies that the limitation of this approach is that the proxy model always depends on the mathematical model.

Deep learning introduces a new era in machine learning, and many unsolved problems in computer science, such as face and voice recognition, have now been partially or entirely solved. Deep learning has a network of multiple layers capable of extracting features from raw data. The depth of architecture is one of the main differences between standard ANN and deep learning networks. In addition, more powerful optimisation functions are used in the training process of deep learning networks. Given the recent progress in activation functions, optimisation algorithms and hyperparameters, deep learning networks have enormous advantages over shallow neural networks (Schmidhuber 2015).

As deep learning has shown significant improvement over other machine learning models in computer science, we investigated the possibility of using this method to improve the efficiency of reservoir simulators. The novelty of this work lies in the use of deep learning to develop a new type of artificial intelligence-based flow forecaster. An advantage of using such a simulator is that it enables us to investigate a wider range of scenarios, and thus, present a more profitable production scheme. First, a dataset was generated based on a mathematical model. Second, a dense multi-layer network was used to relate the inputs and outputs. Third, several case studies were used to verify the feasibility and effectiveness of the developed simulator. Finally, a sensitivity analysis was conducted to examine the effect of each variable

This paper is organised as follows. In Section 2, we describe our proposed methodology to generate the input data and develop a new simulator. In Section 3, we present our results and compare them with the outcomes of a mathematical model followed by a comprehensive sensitivity analysis. Section 4 summarises the findings of this study.

Methodology

The main aim of the proposed approach is to replace the procedure of solving fluid flow equations with a deep learning based network to speed up the simulation. In this way, we elimate the most time-consuming task in the simulation process while significantly reducing the model runtime.

To investigate the feasibility of this approach and understand the effect of each variable, we started from the most basic model: a 1D single-phase oil reservoir. A 1D reservoir simulation has limited applications, such as streamline simulation. Therefore, in an attempt to study a more realistic case, we also considered a 2D gas reservoir. A 2D reservoir simulation can be performed in the preliminary stage of a field development study, when there is only limited data around the well, and provide a good understanding of reservoir resources. Moreover, modelling single-phase gas flow is more complicated than modelling single-phase oil due to the effect of compressibility and the turbulent flow/non-Darcy flow regime.

Since we are dealing with a data-driven model, we required a reliable dataset source. Thus, our model is made of two parts:

- A data generating process to produce data using a mathematical model or commercial simulator, and
- A deep net simulator (DNS) to create an intelligent network (capable of learning and understanding the fluid flow phenomenon) using a deep learning model that would be validated under different scenarios.

Data generating process

1D oil reservoir

In any data-driven model, the quality of the input data is a critical parameter in the accuracy of the final model. This needs to be highlighted, given this study is the first real attempt to create a stand-alone, data-driven reservoir simulator. Therefore, by considering only one dimension, one production well and one type of fluid, we kept the model as simple as possible to better understand the problem. Besides, to avoid neglecting any factors that affect fluid flow and consider all the variables that relate the output to the inputs, we did not use a commercial simulator in the first step. Therefore, to generate the required data, we created a 1D reservoir simulation model by solving the following 1D single-phase flow equations:

$$\begin{aligned}&\frac{\partial}{\partial x}\left[\beta_c\frac{k_xA_x}{\mu_oB_o}\left(\frac{\partial p_o}{\partial x}\right)\right]\Delta x+\frac{\partial}{\partial y}\left[\beta_c\frac{k_yA_y}{\mu_oB_o}\left(\frac{\partial p_o}{\partial y}\right)\right]\Delta y\\&+\frac{\partial}{\partial z}\left[\beta_c\frac{k_zA_z}{\mu_oB_o}\left(\frac{\partial p_o}{\partial z}\right)\right]\Delta z=\frac{V_b}{\alpha_c}\frac{\partial}{\partial t}\left(\frac{\varphi}{B_o}\right)-q_{osc}\end{aligned}\qquad(1)$$

There is a single production well in our system. The developed simulator predicts the pressure of each cell in a 1D single-phase reservoir through different time-steps. Table 1 shows the data range for input variables.

Different attributes were assigned to each cell randomly to create a uniform 1D heterogeneous formation. Then, the created model was solved fully implicitly for two years with a specific time-step. This process was repeated 100 times to set up a dataset containing 5 000 000 data points.

2D gas reservoir

To study the application of deep learning in a more realistic scenario, we used Schlumberger ECLIPSE software to create a 2D gas reservoir model and generate the required dataset. Similar to the 1D oil reservoir case, a fully implicit numerical

Table 1. Range of different variables used to generate 1D datasets
Rb = reservoir barrels. STB = stock tank barrels

Parameters (oil field units)	Min	Max
X-direction cell size (ft)	100	1000
Y-direction cell size (ft)	100	1000
Z-direction cell size (ft)	20	80
Formation volume factor (Rb/STB)	0.8	2
Rock compressibility (1/psi)	3.50E-06	3.50E-06
Permeability (Darcy)	1	60
Porosity (%)	0.04	0.25
Viscosity (cP)	7	16
Well rate (STB/day)	50	300
Time-step (day)	2	30
Number of cells	5	50
Initial pressure (psia)	500	8000

method was used to avoid any convergence problem. We assumed that there was only one production well in the field. Table 2 shows the data range for the input variables. The developed 2D simulator predicts the pressure of each cell through different time-steps.

In all scenarios, the production well was controlled by the bottom hole pressure, and the well rate was constant. In both cases, the models were solved and simulated, and the generated data points were then normalised with zero mean and unit variance before they were used in the deep learning model.

The DNS

The next step was to create a deep learning network. Therefore, it was necessary to divide the dataset into the learning and test datasets. We used the early stopping strategy in the developed network. Therefore, the dataset was divided into three parts: training, validation and test sets. These sets were divided into batches, each containing 500 samples. The most critical section in this research was the network architecture. To find the best network, we started from two dense layers and studied nets up to 25 layers. For each architecture, we optimised the network using different hyperparameters and activation functions. Table 3 shows a brief review of the different parameters we examined in the network architecture.

Validation procedures

Data-driven models are usually validated using a test dataset. Since we created a stand-alone simulator, validating the model through only the use of test dataset was deemed insufficient. To challenge the DNS, we designed 11 examples comprised of six and five configurations for the 1D and 2D reservoirs respectively. Each of these 1D and 2D examples contained 600 and 700 000 data points respectively

Each of these examples were designed carefully to target all the specific parameters a simulator could deal with and challenge the simulator with different rock properties, fluid properties, production settings and simulation constraints. These benchmarks helped us to study the accuracy of the model in each cell within the production time and provide thoughtful feedback to whether the developed simulator could follow the trend of the pressure drop and understand the physics of the problem correctly.

Table 2. Range of different variables used to generate 2D datasets

Parameters (oil field units)	Min	Max
X-, Y-, Z-direction cell size (ft)	20	50
Reservoir depth (ft)	5000	9500
Gas density (lb/ft^3)	0.04	0.048
Initial water saturation (%)	0.16	0.25
Formation volume factor (rb/MSCF)	1.2	13.94
Rock compressibility @ ref pressure (1/psi)	3.00E-06	4.50E-06
Gas viscosity (cP)	0.0124	0.0161
Permeability (md)	30	350
Porosity (%)	0.04	0.25
Well rate (MSCF/d)	1200	10 000
Time-step (days)	10	30
Number of cells in X- and Y-direction	100	4900
Initial pressure (psia)	3000	7500
Bottom hole pressure (psi)	14.7	2000

Table 3. Range of parameters used in the deep-learning network

Parameters	Min	Max
Layers	2	25
Neurons	10	500
Epoch	10	500
Parameters	Methods	
Regularisation	Early stopping, L1, L2, dropout	
Activation functions	Linear, Sigmoid, tanh, ReLU, ELU	
Optimisation function	Stochastic gradient descent (SGD), Momentum, Nesterov's accelerated gradient, Adagrad, Adadelta, RMSprop, Adam, AdaMax	
Loss function	Mean squared error, mean absolute error (MAE), mean absolute percentage error (MAPE), accuracy	

In Table 4, the first three case studies are homogenous reservoirs. Case 4 is a homogeneous reservoir with one cell of low permeability and porosity. It is very critical to study such changes in rock properties because, in reality, we encounter similar changes when there are geology structures, such as faults and anticlines, in an oil field. The last two cases are heterogeneous reservoirs, where permeability and porosity decrease across the reservoir by 10% and 20% respectively.

We solved these case studies using both the developed and mathematical simulators and compared their results. It must be noted that the effect of the remaining properties is presented in the sensitivity analysis section.

Similar to 1D examples, 2D examples were designed to challenge the developed simulator as much as possible. Therefore, all examples in 2D were heterogeneous. To make it more complicated, we considered a mean porosity for each case and applied the normal distribution model to assign porosity over the grids in the reservoir. Then, we used the value of the porosity to calculate the permeability of each cell. Moreover, the initial pressure for each cell was calculated based on its depth.

From case 1 to 5, rock and fluid properties were changed within the range of reasonable values for a typical gas reservoir (Table 5).

Results and discussion

In this research, the developed models were 1D oil and 2D gas reservoirs. In both cases, the dynamic fluid was single phase and produced from one well. The main idea was to create a deep learning model that is capable of learning the physics of the fluid flow and acting like a stand-alone simulator, not just a simple interpolator of the dataset. Unlike the proxy modelling approach that can be used only for its corresponding reservoir, the developed DNS model is a pre-trained, data-driven simulator for a wide range of reservoirs, as specified in Tables 1 and 2. The DNS can instantly predict the real-time reservoir pressure during production in any location across the reservoir as long as its configurations fall in the aforementioned ranges without the need for further rebuilding or training the model. Unlike any conventional reservoir simulator, which requires the properties of all the grid cells to perform the simulation, the DNS prediction relies only on the characteristics of the location (grid cell), distance to the wellbore, initial condition and production settings, without any consideration of the properties of the other

Table 4. Case studies used to validate the developed 1D simulator

Properties	1	2	3	4	5	6
X-direction cell size (ft)	900	700	500	1000	900	900
Y-direction cell size (ft)	700	800	500	800	700	700
Z-direction cell size (ft)	70	70	50	60	70	70
Formation volume factor (Rb/STB)	1	1.2	1.1	1	1	1
Rock compressibility (1/psi)	3.50E-06	3.50E-06	3.50E-06	3.50E-06	3.50E-06	3.50E-06
Permeability (Darcy)	0.025	0.025	0.03	0.03 (0.005)	0.025 (–10%)	0.025 (–20%)
Porosity (%)	20	17	22	22 (1)	20 (–10%)	20 (–20%)
Viscosity (cP)	10	10.5	9	10	10	10
Well rate (STB/day)	150	150	100	200	150	150
Time-step (day)	10	12	10	5	10	10
Production duration (days)	300	300	300	300	300	300
Number of cells	10	10	20	10	10	10
Initial pressure (psia)	5000	5000	6000	5000	5000	5000

Table 5. Case studies used to validate the developed 2D simulator

Properties	1	2	3	4	5
Number of cells in X- and Y-direction	20	40	50	70	70
X-, Y-, Z-cirection cell size (ft)	50	50	20	30	40
Mean porosity (%)	16	15	17	19	20
Reservoir depth (ft)	6000	6700	5300	9000	10000
Rock compressibility @ ref pressure (1/psi)	0.000003	0.000003	0.0000035	0.0000035	0.0000035
Gas density (lb/ft^3)	0.04	0.04	0.042	0.043	0.03
Initial water saturation (%)	18	16	20	23	20
Well rate (MSCF/d)	2000	200	500	800	1300
Bottom hole pressure (psia)	1200	1100	1300	1250	1150
Time-step (days)	10	10	10	10	20
Production duration (days)	100	300	100	500	1200

cells. With this approach, we can simulate any part of the reservoir without the need to simulate the whole reservoir. This strategy allowed us to speed up the reservoir simulation significantly, meaning it took only a fraction of a second to predict the pressure for a specific location.

1D oil reservoir

Using the data generated for this scenario, we created a deep learning based network, which was optimised, starting from two layers to up to 25 layers (Fig. 1). Given that MAPE values of the developed models, containing seven layers or less, were more than 30%, the MAPE value for these six models were not included in this chart. According to this figure, the network with 15 layers provides the best results, while increasing the number of layers to more than 15 did not provide any significant advantages. Table 6 provides the final network architecture used in this study.

The comparison of accuracy among training, validation and test datasets are shown in Table 7. The low MAPE value (14.3–15.1) indicates that the developed simulator could extract the existing patterns among the data points and obtain adequate understanding between the inputs and desired output.

In addition to comparing the results of the proposed model using R^2 and MAPE of the test dataset, six reservoir models (Table 4) were simulated with the developed and numerical simulators, and the results are compared in Table 8. As it can be seen, the values of R^2 are between 0.95 to 0.99, and the values of MAPE for all the cases are less than 5%, which validates the reliability of the developed approach.

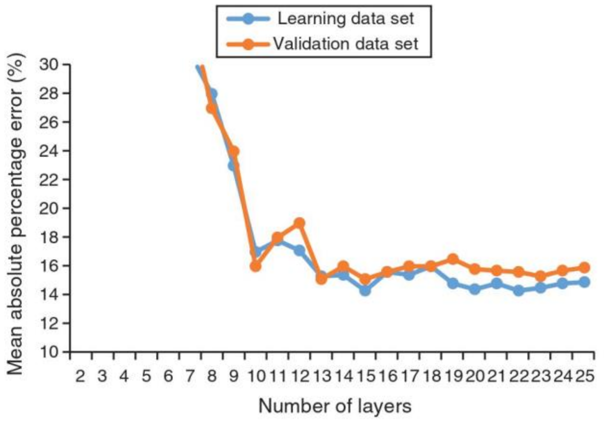


Fig. 1. Effect of using a different number of layers on the accuracy of the learning and validation datasets.

Table 6. Network architecture for 1D simulator

Parameters	Values
Layers	15
Neurons	5 to 100
Epoch	200
Regularisation	Early stopping, dropout
Activation functions	Sigmoid, tanh, ReLU
Optimisation function	AdaMax
Loss function	MAPE

In cases 1 and 2, all parameters were selected so that these values were placed in the middle range of the domain. The developed simulator generated excellent results in both cases, with R^2 of 0.99 and 0.98 and MAPE of 1.55% and 3.72%, respectively.

In case 3, the grid size was increased to see whether the developed simulator could solve the problems with a high number of grids. In case 4, the effect of sudden low transmissibility was studied. High values of R^2 and low values of MAPE in both cases confirm the capability of the developed simulator.

In cases 5 and 6, permeability and porosity changed across the reservoir by 10% and 20% respectively. Even with these heterogeneities, the developed simulator was able to predict the cells' pressure within an acceptable margin of error. The MAPE and R^2 for case 5 was 4.78% and 0.95, and 4.02% and 0.95, for case 6.

To analyse the simulator performance, we conducted a range of sensitivity analyses for the different parameters. This analysis allowed us to determine whether enough training data had been presented for a specified range.

We considered example 1 in Table 4 as the base model. The following parameters were evaluated:

- Initial pressure,
- Permeability,
- Grid size, and
- Well rate.

Figs 2–5 compare the results of the developed simulator to a numerical one as these parameters changed. According to the results, the DNS could predict the cell pressure for various reservoir conditions and settings within an acceptable margin of error. Variation of accuracy in these results helped us to understand how we can improve the training dataset, and consequently, the developed model.

Table 7. Comparing the accuracy of the predicted cell pressure for different datasets in the 1D oil reservoir scenarios

	Training dataset	Validation dataset	Test dataset
MAPE (%)	14.3	15.1	14.9
MAE (psi)	28.9	29.6	29.7
R^2	85.0	84.1	84.3

Table 8. Comparing the accuracy of the predicted cell pressure for different examples in the 1D oil reservoir scenarios

	1	2	3	4	5	6
MAPE (%)	1.55	3.72	2.27	1.43	4.77	4.02
MAE (psi)	5.23	8.97	6.75	4.99	11.26	10.81
R^2	0.99	0.98	0.97	0.99	0.95	0.95

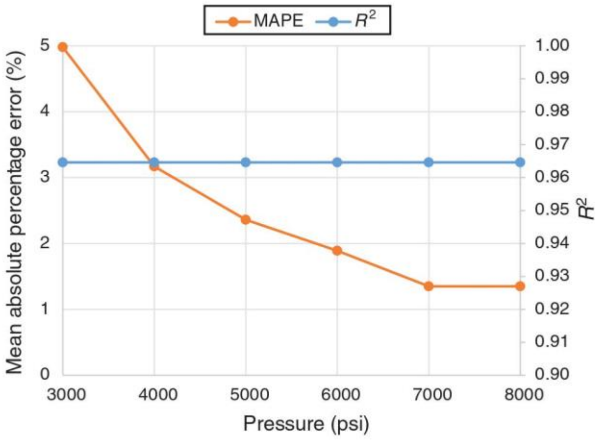


Fig. 2. Effect of pressure on the accuracy of the developed simulator.

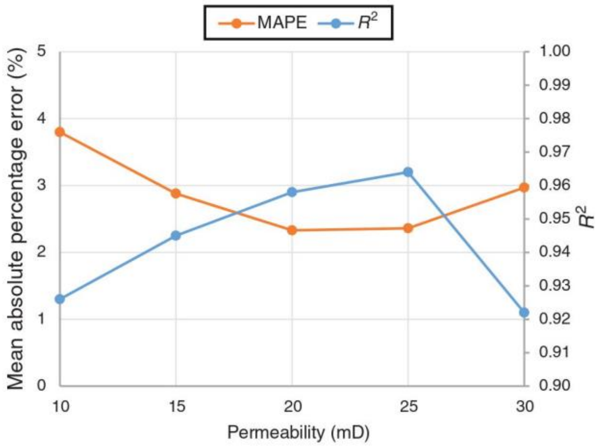


Fig. 3. Effect of permeability on the accuracy of the developed simulator.

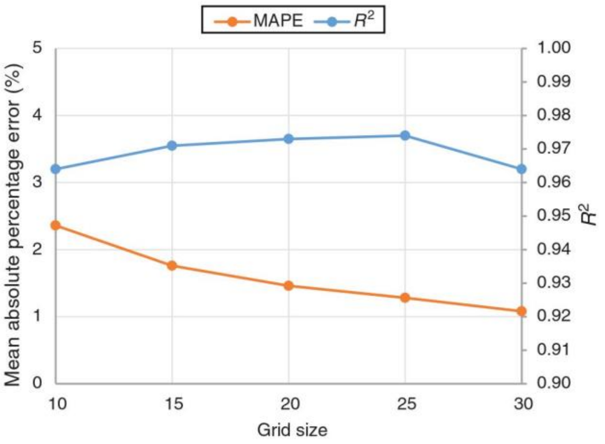


Fig. 4. Effect of the number of cells on the accuracy of the developed simulator.

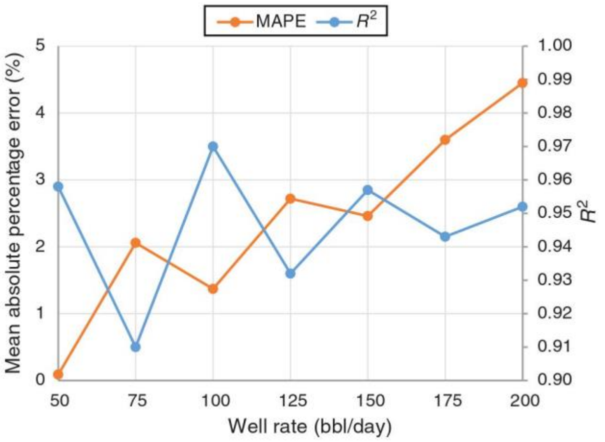


Fig. 5. Effect of well rate on the accuracy of the developed simulator.

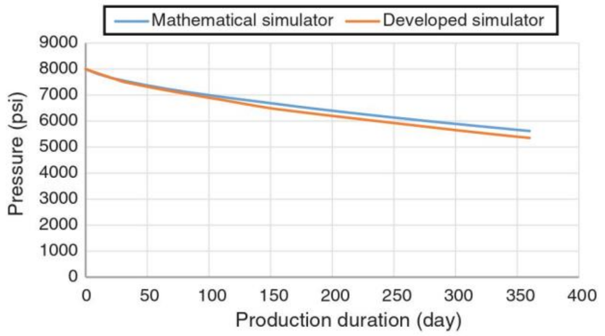


Fig. 6. Pressure drop during production in the cell connected to the wellbore.

In addition to the accuracy, a reliable simulator must have a good understanding of the problem and its output must follow a desirable trend.

There are always inevitable errors between the outputs of mathematical simulations and the field production data. This difference is due to the uncertainty in the inputs, which introduce some errors to the results. In reservoir engineering, we use the observed pressure and production data to calibrate the simulated model. This data is used to reproduce the past behaviour of the reservoir. This 'history matching' process is very time-consuming. After history matching the model, the calibrated model can be used to predict the future behaviour of the reservoir with a higher degree of confidence.

Therefore, any created model must be accurate, and at the same time, follow a desirable trend. In other words, it must fully replicate the fluid behaviour in porous media. Figs 6 and 7 show the pressure of two cells in a reservoir with an initial pressure of 8000 psi; one cell contains the wellbore, and the other cell is its neighbour. These two cells had the highest pressure drop during oil production. Although, as can be seen in Figs 6 and 7, there are noticeable mismatches between the output of the DNS and the numerical model; the pressure profile tended to decrease with a similar slope to the mathematical and data-driven model. This shows that it is

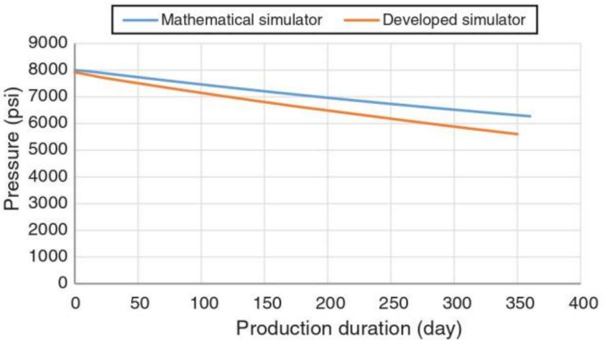


Fig. 7. Pressure drop during production in the cell adjacent to the cell connected to the wellbore.

Table 9. Network architecture of the 2D simulator

Parameters	Values
Layers	9
Neurons	2–150
Epoch	20 000
Regularisation	Early stopping, dropout
Activation functions	Sigmoid, linear, ReLU
Optimisation function	Adagrad
Loss function	MAPE

possible to adjust the uncertainty of the developed model and improve its accuracy with history matching, similarily to the history matching process in a mathematical-derived model.

Although the accuracy of our model can be improved, it should be noted that 1D simulations are not the final objective of this ongoing research. Regardless that the results of the 1D simulator are not remarkable, our study of reservoir simulation in 1D allowed us to grasp a more in-depth understanding of the problem, allowing for future investigation of more complex scenarios.

2D gas reservoir

Results of the developed model for a 1D oil reservoir show that the idea of using deep learning to develop a simulator in 2D was worthy of investigation. To do so, we used the generated data and taught the physics of 2D fluid flow to a nine-layer network. Table 9 provides the network architecture used in this model. In this case, the results were even more accurate. MAPE for training, validating and test data were 0.82%, 0.82% and 0.84% respectively. Mean absolute error (MAE) for these sets were 19.07, 19.17 and 19.10 psi respectively, and R^2 in all sets were almost 1.

Similar to the 1D case, we evaluated the developed simulator with a couple of benchmarks, which are shown in Table 5. These reservoirs were simulated with both ECLIPSE and the developed DNS. As can be seen in Table 10, the low values of MAPE and MAE and the high values of R^2 indicate that, even in such a complicated case, the developed simulator could predict the pressure drop almost flawlessly. Even though the values of variables in each benchmark were very different and covered a wide range of values, the developed simulator could model all of them accurately.

Table 10. Comparing the accuracy of the predicted cell pressure for different sets in the 2D gas reservoir scenarios

	1	2	3	4	5
MAPE (%)	1.54	2.38	1.85	0.55	0.76
MAE (psi)	40.94	75.81	42.97	22.38	33.77
R^2	0.99	0.88	0.99	0.99	0.99

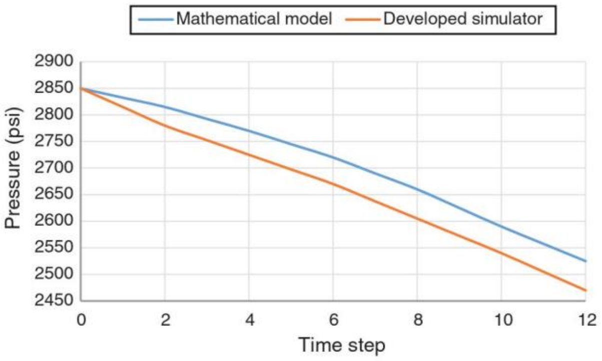


Fig. 8. Comparison of the pressure in the cell (15,15) in benchmark 1.

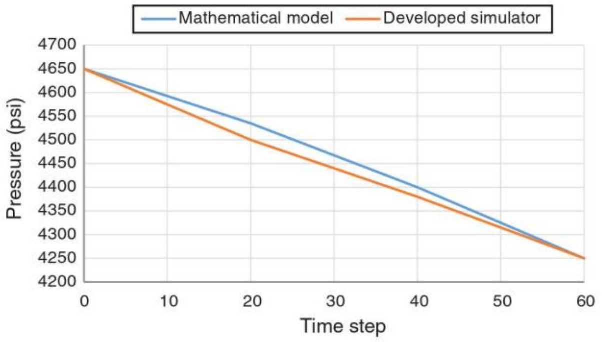


Fig. 9. Comparison of the pressure in the cell (5,70) in benchmark 5.

As the grid size increased from benchmark 1–5, similar to the 1D case, the margin of error decreased. In these benchmarks, low and high well rates were studied. Even though the minimum well rate in the training data was 2000 MSCF, by considering values lower than this, we investigated whether the simulator understands the pattern correctly and extrapolated the values beyond the input's domain. Results confirmed that the developed simulator could estimate the output accurately, despite these inputs.

In addition to Table 10, Figs 8 and 9 show the capabilities of the developed simulator to predict the pressure drop of a cell over the production time. In these figures, the blue line is the output calculated with the numerical simulator, and the red line indicates the values from the developed simulator. Fig. 8 shows the pressure changes in one of the cells (x = 15,

$y = 15$) in benchmark 1. As can be seen in this figure, through the 12 time-steps, the trend and slope of the lines are the almost the same, except in the earlier time-steps. Similarly, Fig. 9 shows the pressure change in another cell ($x = 5$, $y = 70$) over 60 time-steps. Similar to the other cell, in this one, the predicted values from the DNS matched the values calculated with the mathematical model.

Conclusions

In this study, the possibility of replacing fluid flow equations with a data-driven network was explored. Based on this approach, an optimised stand-alone reservoir simulator for 1D and 2D single-phase flow in subsurface reservoirs was developed. The developed model shows promising results in estimating the pressure drop throughout the reservoir using the same inputs as a physics-based simulator. Unlike a proxy model for a specific reservoir, a stand-alone reservoir simulator was developed to cover a wide range of input data. For studied examples, the MAPE of the proposed model was less than 4% for 1D simulation and less than 2.4% for a 2D simulation. In the 1D scenario, we applied sensitivity analysis to study the effect of initial pressure, permeability distribution, grid size and well rate. This allowed us to improve the DNS by broadening our understanding of the range of data used throughout the training of the model. Comparing the outcome of the DNS with the numerical results (MAPE < 5%) confirmed the capacity of this approach. The outcome of this investigation suggests that the DNS has the potential to become a fast alternative to conventional reservoir simulation, leading to the exploration of more scenarios in history matching and field development planning. A more complex scenario is currently in progress and will be presented in future publications.

Conflicts of interest

The authors confirm that there are no known conflicts of interest associated with this publication.

Acknowledgments

The authors are grateful to Alexander Arthurson for his assistance and Santos Ltd for their financial support in this research.

Reference

Alenezi, F., and Mohaghegh, S. (2017). Developing a Smart Proxy for the SACROC Water-Flooding Numerical Reservoir Simulation Model. In 'SPE Western Regional Meeting, 23–27 April, Bakersfield, California.' (Society of Petroleum Engineers). doi:10.2118/185691-MS

Artun, E., Ertekin, T., Watson, R., and Al-Wadhahi, M. (2011). Development of universal proxy models for screening and optimization of cyclic pressure pulsing in naturally fractured reservoirs. *Journal of Natural Gas Science and Engineering* **3**, 667–686. doi:10.1016/j.jngse.2011.07.016

Bangerth, W., Klie, H., Wheeler, M. F., Stoffa, P. L., and Sen, M. K. (2006). On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences* **10**, 303–319. doi:10.1007/s10596-006-9025-7

Beckner, B. L., Haugen, K. B., Maliassov, S., Dyadechko, V., and Wiegand, K. D. (2015). General Parallel Reservoir Simulation. In 'Abu Dhabi International Petroleum Exhibition and Conference, 9–12 November, Abu Dhabi, UAE.' (Society of Petroleum Engineers). doi:10.2118/177532-MS

Cardoso, M. A., and Durlofsky, J. L. (2010). Use of Reduced-Order Modeling Procedures for Production Optimization. *SPE Journal* **15**, 426–435. doi:10.2118/119057-PA

Ghassemzadeh, S., and Charkhi, A. H. (2016). Optimization of integrated production system using advanced proxy based models: A new approach. *Journal of Natural Gas Science and Engineering* **35**, 89–96. doi:10.1016/j.jngse.2016.08.045

Ghassemzadeh, S., and Pourafshary, P. (2015). Development of an intelligent economic model to optimize the initiation time of gas lift operation. *Journal of Petroleum Exploration and Production Technology* **5**, 315–320. doi:10.1007/s13202-014-0140-z

Ghassemzadeh, S., Shafflie, M., Sarrafi, A., and Ranjbar, M. (2013). The Importance of Normalization in Predicting Dew Point Pressure by ANFIS. *Petroleum Science and Technology* **31**, 1040–1047. doi:10.1080/10916466.2011.598895

Ghassemzadeh, S., Schaffie, M., Sarrafi, A., and Ranjbar, M. (2014). Predicting Dew Point Pressure: Using a Hybrid Intelligent Network. *Petroleum Science and Technology* **32**, 2969–2975. doi:10.1080/10916466.2014.919004

Goodwin, N. (2015). Bridging the Gap Between Deterministic and Probabilistic Uncertainty Quantification Using Advanced Proxy Based Methods. In 'SPE Reservoir Simulation Symposium, 23–25 February, Houston, Texas, USA.' (Society of Petroleum Engineers). doi:10.2118/173301-MS

Knudsen, B. R., and Foss, B. (2015). Designing Shale-Well Proxy Models for Field Development and Production Optimization Problems. *Journal of Natural Gas Science and Engineering* **27**, 504–514. doi:10.1016/j.jngse.2015.08.005

Li, H., and Zhang, D. (2007). Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods. *Water Resources Research* **43**, W09409. doi:10.1029/2006WR005673

Mohaghegh, S. D., Gaskari, R., and Maysami, M. (2017). Shale Analytics: Making Production and Operational Decisions Based on Facts: A Case Study in Marcellus Shale. In 'SPE Hydraulic Fracturing Technology Conference and Exhibition, 24–26 January, The Woodlands, Texas, USA.' (Society of Petroleum Engineers). doi:10.2118/184822-MS

Sbiga, H. M., and Potter, D. K. (2017). Prediction of Resistivity Index by Use of Neural Networks With Different Combinations of Wireline Logs and Minimal Core Data. *SPE Reservoir Evaluation & Engineering* **20**, 240–250. doi:10.2118/181751-PA

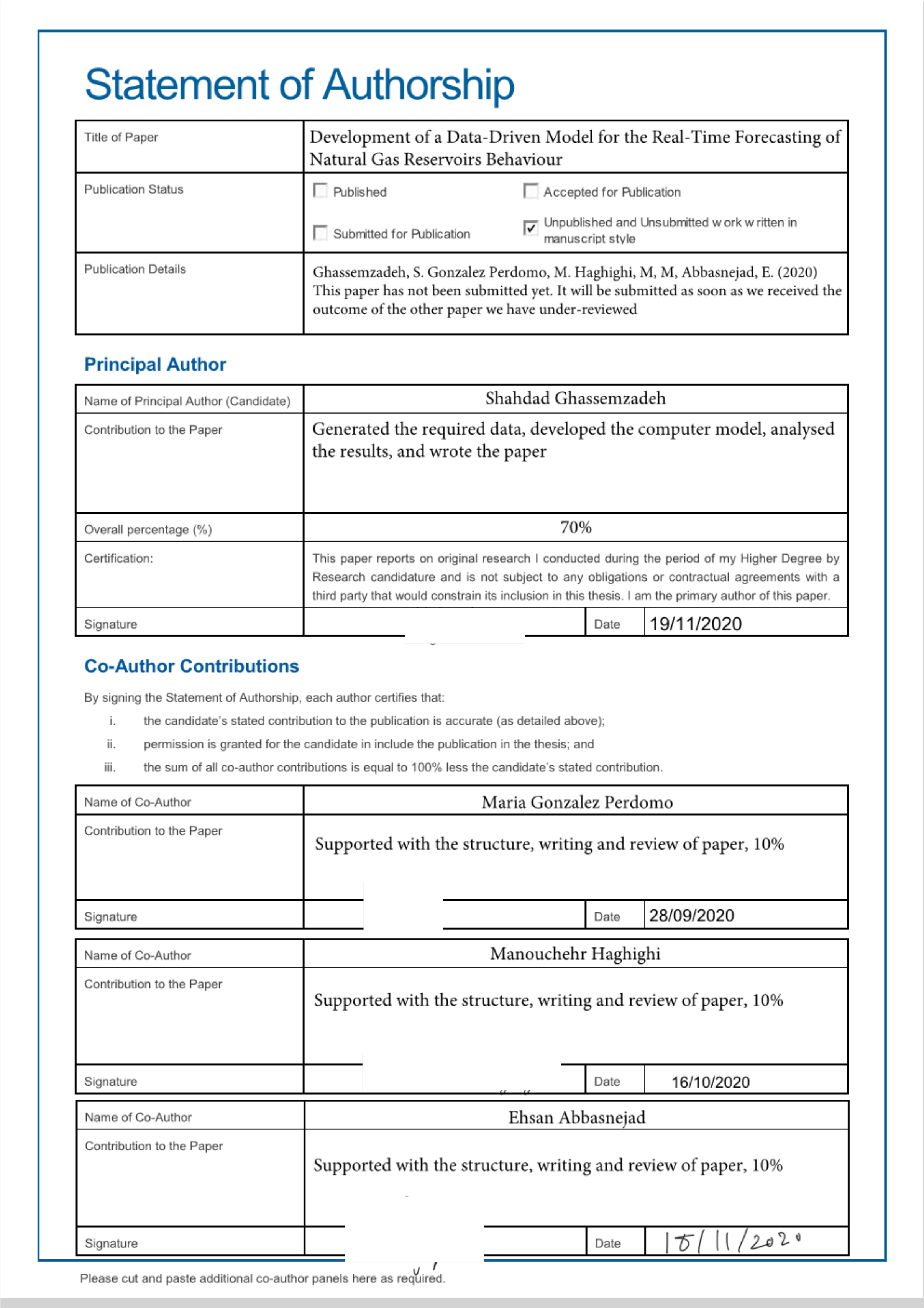
Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks* **61**, 85–117. doi:10.1016/j.neunet.2014.09.003

Shams, R., Esmaili, S., Rashid, S., and Suleymani, M. (2015). An intelligent modeling approach for prediction of thermal conductivity of CO_2. *Journal of Natural Gas Science and Engineering* **27**, 138–150. doi:10.1016/j.jngse.2015.08.050

van Doren, J. F. M., Markovinović, R., and Jansen, J.-D. (2006). Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Computational Geosciences* **10**, 137–158. doi:10.1007/s10596-005-9014-2

Wood, D. A. (2014). Optimization applications seeking to improve performance of natural gas reservoirs, production, processing and utilization: A selection of case studies. *Journal of Natural Gas Science and Engineering* **21**, A1–A3. doi:10.1016/j.jngse.2014.10.012

4. Development a Data-Driven Model for the Real-Time Forecasting of Natural Gas Reservoirs’ Behaviour



Statement of Authorship

Title of Paper	Development of a Data-Driven Model for the Real-Time Forecasting of Natural Gas Reservoirs Behaviour
Publication Status	□ Published □ Accepted for Publication □ Submitted for Publication ☑ Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	Ghassemzadeh, S. Gonzalez Perdomo, M. Haghighi, M, M, Abbasnejad, E. (2020) This paper has not been submitted yet. It will be submitted as soon as we received the outcome of the other paper we have under-reviewed

Principal Author

Name of Principal Author (Candidate)	Shahdad Ghassemzadeh		
Contribution to the Paper	Generated the required data, developed the computer model, analysed the results, and wrote the paper		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	19/11/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate’s stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate’s stated contribution.

Name of Co-Author	Maria Gonzalez Perdomo		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	28/09/2020

Name of Co-Author	Manouchehr Haghighi		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	16/10/2020

Name of Co-Author	Ehsan Abbasnejad		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	15/11/2020

Please cut and paste additional co-author panels here as required.

Abstract

Reservoir simulation is an area within reservoir engineering where computer models are used to predict the flow of fluids through porous media. Reservoir simulation is typically used to quantify the uncertainties and to assist the decision-making process within the petroleum industry. The first step in achieving this is through the calibration of the model through a history matching process via the use of production data, depending on its availability. Therefore, during the reservoir management study and field development planning, the desired reservoir is must be simulated multiple times in order to investigate a range of different parameters and operation strategies. Due to the relatively long runtime that each simulation requires, the overall duration of the reservoir management study and field development planning is extremely time intensive. Therefore, engineers are always looking for a workflow to reduce the simulation time and, consequently, to speed up the decision-making process. Through the use of deep learning, a machine-learned model is developed in this study, assisting the time-consuming partial differential diffusivity equations to be formulated explicitly, while keeping the accuracy found through the implicit approach. With this approach, a data-driven simulator is developed to predict the pressure across a drainage area of a vertical well in a dry gas reservoir instantly. This simulator is a pre-trained deep learning model and can represent a wide range of rock and fluid properties to cover a large number of reservoirs unlike conventional proxy models, which represent only one corresponding reservoir. All the data required for training, testing and validation are generated by a commercial reservoir simulation software (ECLIPSE). Unlike the conventional proxy models, which must be rebuilt for a new reservoir, the developed simulator does not need to be retrained or rebuilt for any new reservoir as long as the reservoir configurations lie within the training data specified in this study. In this research, the deep net simulator (DNS), the developed model, showed a remarkable accuracy for the training, validation, and test sets. All values are in a similar range (Mean Absolute Percentage Error = 0.9, Mean Absolute Error = 11.0, Mean Relative Error = 0.0, Mean Squared Error = 651.3, Root Mean Squared Error =25.5, and R-Squared = 1). To investigate DNS even further, we designed multiple benchmarks, which consist of a wide range of initial conditions, geological conditions, and production settings. The average observed error in these benchmark cases, where the reservoir configuration lay within the specified training range, was very close to zero. This low error means that DNS can show extremely accurate results when the reservoir configuration lies within the specified training database range.

1. Introduction

Reservoir simulation is a crucial asset, used to achieve an adequate understanding and interpretation of reservoir behaviour in reservoir management studies. Additionally, reservoir simulation is used to create the fluid production profile of a reservoir. By incorporating this profile with the respective oil and gas price forecasts, engineers can generate cash flow predictions that can be used later to quantify the financial outcome for any given scenario.

Reservoir simulation is typically used for the following: to quantify reservoir uncertainties, calibrate the model through history matching, and to optimise the field production. Each of these respective tasks would generally require the completion of hundreds, or even thousands of simulations.

Furthermore, to automate and accelerate the real-time decision-making process and manage a reservoir efficiently during its production life, we are required to update the model continually. These updates are associated with real-time data, gathered from respective real-time measurements of various time-dependent properties. Therefore, it can be very advantageous to have an accurate and fast workflow that is capable of being updated rapidly once presented with new data (Artun 2017).

Such a workflow could potentially be obtained by relying on machine learning techniques. These techniques could allow us to develop a data-driven model, capable of reproducing the relationship that exists between the input and output variables during the training process. The key difference between data-driven and physics-based models is the presence of an accepted visible correlation that describes the natural phenomena in physics-based models. Unlike physics-based models, data-driven models do not assume such a relationship and capture knowledge about the phenomena only through the extraction of features existing in the observed data (Kulga, Artun et al. 2017).

Regardless of how impractical this may appear, this feature makes data-driven models an incredibly fast tool in regression problems, an example being in the prediction of reservoir behaviour. The reason is that a data-driven model creates an explicit numerical correlation approximate to the desired output, in contrast with the conventional mathematical approach, which requires many iterations to provide a proper approximation of the output.

Given the above characteristics of data-driven models, the advantages of this approach can become more apparent in the following situations:

- When there are no reliable physics-based models available
- When there is a time limit, and the runtime of a physics-based model is significantly longer than expected
- When not all the data required is available for the physics-based model to use.

There is a high possibility that at least one of these aforementioned situations will be present throughout a reservoir simulation study (Kulga, Artun et al. 2018).

The introduction of powerful hardware and software in recent years has sped up reservoir simulations. However, tasks such as history matching and uncertainty quantification, computational inefficiency and long runtimes remain an ongoing problem in this area. Even though the use of high-speed computers can reduce the runtime, there is a considerable associated cost (Eldred, Orangi et al. 2014, Beckner, Haugen et al. 2015). Since we require the completion of hundreds, or even thousands, of simulations for any of the previously-mentioned tasks (history matching, uncertainty quantification and optimisation), seeking a faster alternative is inevitable, especially if we require an efficient yet fast workflow for the reservoir management study.

To overcome this inefficiency, researchers have frequently opted to create a proxy model of a reservoir using mathematical or machine learning methods in an attempt to discover faster workflow techniques. Reduced-order modelling methods, such balanced truncation, proper orthogonal decomposition, trajectory-piecewise linear and dynamic mode decomposition, have been widely used to reduce the dimensions of the problem in order to create a fast approximate numerical solution (Antonio Cardoso and J. Durlofsky 2010, Kaleta, Hanea et al. 2011, Ghasemi, Yang et al. 2015, He, Xie et al. 2016).

In addition to reduced-order modelling, machine learning techniques, specifically using a shallow artificial neural network (fewer than 3 hidden layers), have been used to construct proxy models for different types of reservoirs (Artun, Ertekin et al. 2011, Goodwin 2015, Kalantari-Dahaghi, Mohaghegh et al. 2015, Alenezi and Mohaghegh 2016, Ghassemzadeh and Charkhi 2016, He, Xie et al. 2016, Alenezi and Mohaghegh 2017, Chen, He et al. 2017, Mohaghegh, Gaskari et al. 2017, Kim and Shin 2018, Nwachukwu, Jeong et al. 2018).

Even though these methods provide robust results, each proxy developed with these approaches can be used only for one corresponding reservoir. This means that for a completely new reservoir, the construction of a proxy model must be restarted from scratch. Furthermore, considering the time that is spent building, validating, and running the proper proxy model of a reservoir, in some cases, running a numerical reservoir simulation iteratively could be found to be more time-efficient in comparison.

The aim of this article is to continue ongoing research into the application of deep learning for reservoir simulation (Ghassemzadeh, Perdomo et al. 2019) and to address the practical possibility of using a data-driven simulator, as opposed to a numerical physics-based simulator, for a natural gas reservoir. This developed model will be used as a forecasting tool to predict the pressure at any point within the reservoir instantly, given the initial conditions, operation parameters and reservoir characteristics. This approach can be interpreted as a proxy to the numerical reservoir simulator to represent multiple reservoirs, in comparison with a conventional proxy modelling approach, in which it can only be used to represent its corresponding reservoir.

2. Methodology

The steps that were taken to develop this proposed model are in some way similar to the steps previous researchers have proposed in their development of a conventional proxy model representing a specified reservoir. However, unlike for the convention proxy modelling approach, the database that was used in the training of our model was collected from a range of different reservoir models, all with diverse characteristics, descriptions, and settings. This approach results in a complex database covering a range of complexities that may occur within a reservoir, Figure 1. This approach can be used to help us create more sophisticated, complex data-driven models that can be used as a proxy for various numerical reservoir simulations.

The main challenge throughout this study was to train such complicated problems and to extract the complex features amongst this big data. We achieved this by relying on one of the latest machine learning techniques, deep learning.

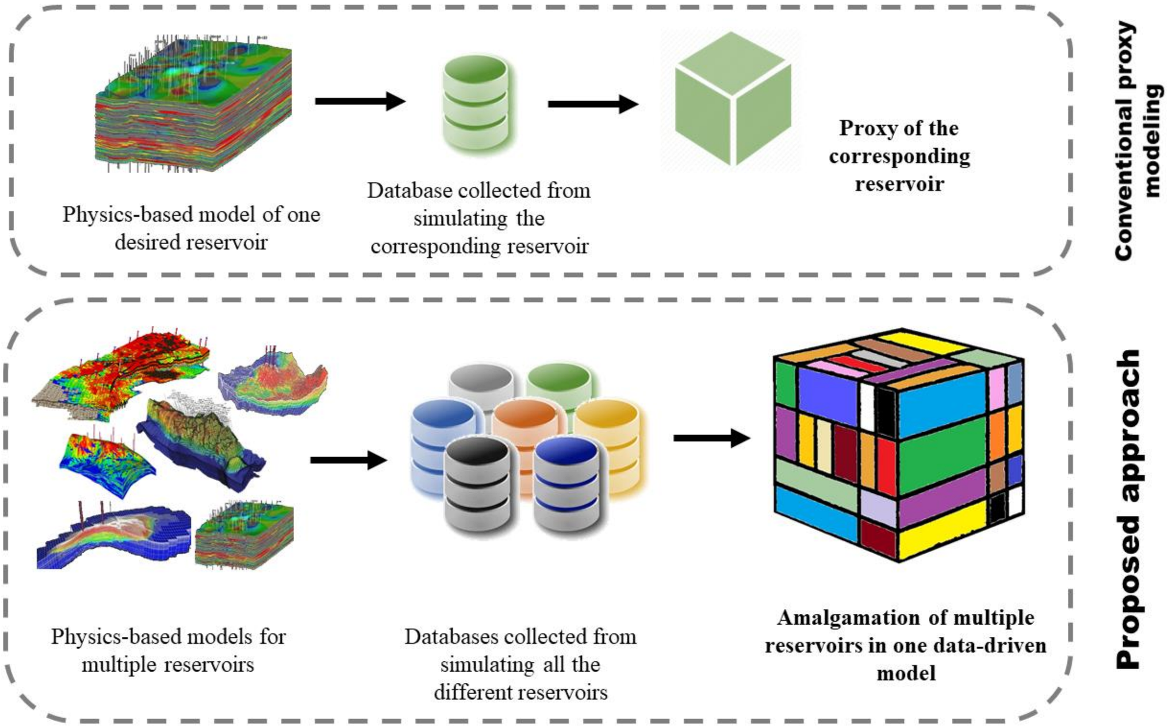

Physics-based model of one desired reservoir
Database collected from simulating the corresponding reservoir
Proxy of the corresponding reservoir
Conventional proxy modeling
Physics-based models for multiple reservoirs
Databases collected from simulating all the different reservoirs
Amalgamation of multiple reservoirs in one data-driven model
Proposed approach

Figure 1, Flowchart of building DNS. This Figure demonstrates the difference between a traditional proxy model and a DNS model. While a proxy model corresponds to a single desired reservoir, our DNS can be used as a proxy for a wide variety of reservoirs.

2.1.The development of a Spatiotemporal database

During this step, we generated a spatial-temporal database to train a deep learning model and, consequently, to replicate the behaviour of a commercial simulator. Future work in this study involves the use of real field data, to fine-tune our developed model’s potential to match the reality of the reservoir behaviour. This allows us to reduce the uncertainty present within diffusivity equations. As there are different physical phenomenon used to govern different reservoirs, the first step in developing an alternative tool to reservoir simulation is to target a specific reservoir type.

During this study, we have concentrated on natural gas reservoirs with gas expansion as the primary drive mechanism. There are serious reasons to focus on gas reservoirs. First, from a sustainability point of view, many consider natural gas as a clean bridge to a renewable future (Brown, Krupnick et al. 2009, Kerr 2010, Tour, Kittrell et al. 2010, Leung 2015). Given how many natural gas resources are prospective, we prioritised gas reservoirs over oil resources. Furthermore, in comparison with single-phase oil flow, single-phase gas flow is more complicated, and therefore more challenging to model using a data-driven model. There are two main differences between the single-phase flow of oil and gas. Firstly, gas flow velocities, in general, are higher to the extent that the inertial forces may become significant and no longer be ignored. Therefore, Darcy's Law for flow in porous media should be replaced by the more complex Law of Forchheimer, which includes the effect of inertial forces. Secondly, gas compressibility and gas viscosity are dependent on pressure, and this results in a differential equation for gas flow that is substantially non-linear (Hagoort 1988).

Since single-phase gas flow is investigated in this research, we selected a black oil model, which is accurate enough to formulate and represent the behaviour of gas reservoirs (Iwere, Moreno et al. 2006). Furthermore, the model we considered to simulate the fluid flow in the dry gas reservoir is a 1-layer (2-dimensional) system with rectangular and uniform grid cell distribution. We assumed the drainage area is partially saturated with gas, whilst the water-phase is considered immobile.

We used a commercial reservoir simulator (ECLIPSE) to develop the spatial-temporal data of the gas reservoirs. Using this simulator, we built a template black oil model that represented the drainage area of a vertical well in a dry gas reservoir. Based on a predetermined range of reservoir configurations, we then generated the required spatial-temporal database.

In the development of our model, we first began with the random generation of 50 synthetic reservoir configurations. These models were designed with a random generator, which randomly chooses values that lie within the pre-specified range of data presented in Table 1. These values are based on real field data obtained from various reservoirs located around the globe (Zou, Zhu et al. 2012). By using randomised realistic values, we aimed to generate synthetic reservoir configurations with various properties that would allow us to determine and describe the physical behaviour of gas flow in porous media.

The first step in each iteration of this flowchart was to create an ECLIPSE data file. ECLIPSE data files comprise the necessary reservoir descriptions and settings to model a reservoir. To create the proper data file, the required values were read randomly from the range presented in Table 1 and written into a file in the data file format. It should be noted that each value obtained from each row in Table 1 was chosen independently from each value in other rows in order to maximise the randomness.

To make this model more realistic, we specified that the geology in our models was heterogeneous in every case. This allowed us to consider the specified mean porosity for each reservoir by applying the normal distribution function, allowing us to assign the porosity values over the grids in the reservoir. In order to calculate the absolute permeability for each cell, we used one of the most common modifications of the Wyllie-Rose empirical equation, Timur’s equation (Timur 1968). This equation, as shown below, provides a reasonable estimation of the absolute permeability of sandstone formations (Schön 2015):

$$\mathrm{k} = \left[100 \times \frac{\varphi^{2.25}}{\mathrm{S_{w,irr}}}\right]^2 \qquad \text{Equation 1}$$

where the $\mathrm{S_{w,irr}}$ is the irreducible water saturation and is considered to be 20 % for all cases (Baker, Yarranton et al. 2015). This was followed by the calculation of the reservoir’s initial pressure. To do this, we used hydrostatic pressure based on the assigned reservoir depth (Hagoort 1988). This is given by:

$\mathrm{P_i} = 0.475 \times \mathrm{Depth}$ Equation 2

Apart from the previously mentioned variables, some features and settings were fixed throughout all the scenarios and models. In this study, we considered that the production well was controlled by the bottom-hole pressure (BHP). This is a practical setting in a gas field, as it helps us to avoid sand production and water coning, and satisfies the constraints in regards to production facilities and regulatory authorities (Guo and Ghalambor 2014). Therefore, the production well present in each model was produced at its assigned well rate until it reaches its allocated BHP. It should be noted that both the saturation and relative permeability values were kept constant; this is due to this study being focused on single-phase flow.

After generating the data files, the ECLIPSE software is used to simulate these data files. A parsing algorithm is then used to read and convert the required values from the specified data files, and their corresponding output files are converted into tabular-format files. Finally, these tabular-format files are then combined into a single file, resulting in a database of approximately 20,000,000 data points, corresponding to all 50 synthetic models.

The training data files included:

Inputs:

- Time
- Cell size in X, Y, Z – direction
- Formation volume factor for water and gas
- Viscosity
- Porosity
- Relative permeability in X, Y- directions
- Gas density
- Initial water saturation
- Rock compressibility
- Reservoir depth
- Well location
- Well rate
- Initial pressure
- Bottom hole pressure

Output:

- Cell pressure

In order to help the deep learning model understand the dynamic concept of this problem, we included time as one of the input data sets in the model. To speed up the learning process, we normalised the input values with zero mean and unit variance.

Table 1, Range of different variables used to develop DNS

Parameters (Oil Field Units)	**Min**	**Max**
Grid Size	20 x 20	70 x 70
Z - Direction Cell Number	1	1
X, Y, Z - Direction Cell Size	50	300
Gas Density (lb/ft^3)	0.04	0.048
Initial Water Saturation (%)	0.16	0.25
Rock Compressibility @ 3500 psi (1/psi)	3.00E-06	3.50E-06
Mean Porosity (%)	15	25
Permeability (mD)	Timur's empirical equation	
Well rate (MSCF/d)	12000	800000
Reservoir Depth (Top) (ft)	5000	9000
Initial pressure (psia)	0.475 * Depth	
Bottom Hole Pressure (psia)	800	1300
Time step (days)	10	30
Production Time (days)	1000	4500

2.2.The training of the model

The next step is the determination of a suitable architecture and the selection of a proper training configuration for the mentioned architecture. The database that was generated in the previous step is then fed into this architecture, such that the trained model can predict the same results as a numerical simulator.

We used dense-layered deep learning and attempted different architectures to create the best possible model. We started with an architecture that contains one hidden layer and used a variety of different combinations of neurons, activation functions, optimisation functions and loss functions. Once we achieved the best possible parameters within the architecture with one hidden layer, we increased the number of hidden layers to two and attempted to optimise the new architecture by testing new values for these hyper-parameters. We continued to increase the layers until we were assured that there was no chance to improve the accuracy of the model by increasing the layers. We used the early stopping strategy with a validation set using the mean absolute percentage error (MAPE) as the monitoring metric in the training process to avoid the overfitting problem. Table 2 shows the different values we attempted for each parameter and hyperparameter to create the proposed model. Figure 2 illustrates the schematic diagram of the deep learning topology we used in this study.

Table 2, Parameters and hyper-parameters used in the deep learning network

Parameters	Values
Layers	Up to 25
Neurons	2 - 300
Epoch	5000
Regularisation	Drop out
Activation functions	Linear, Sigmoid, tanh, ELU, Relu
Optimisation Function	ADAgrad, ADAselta, ADAM, SGD
Loss Function	MSE, MAPE, MAE

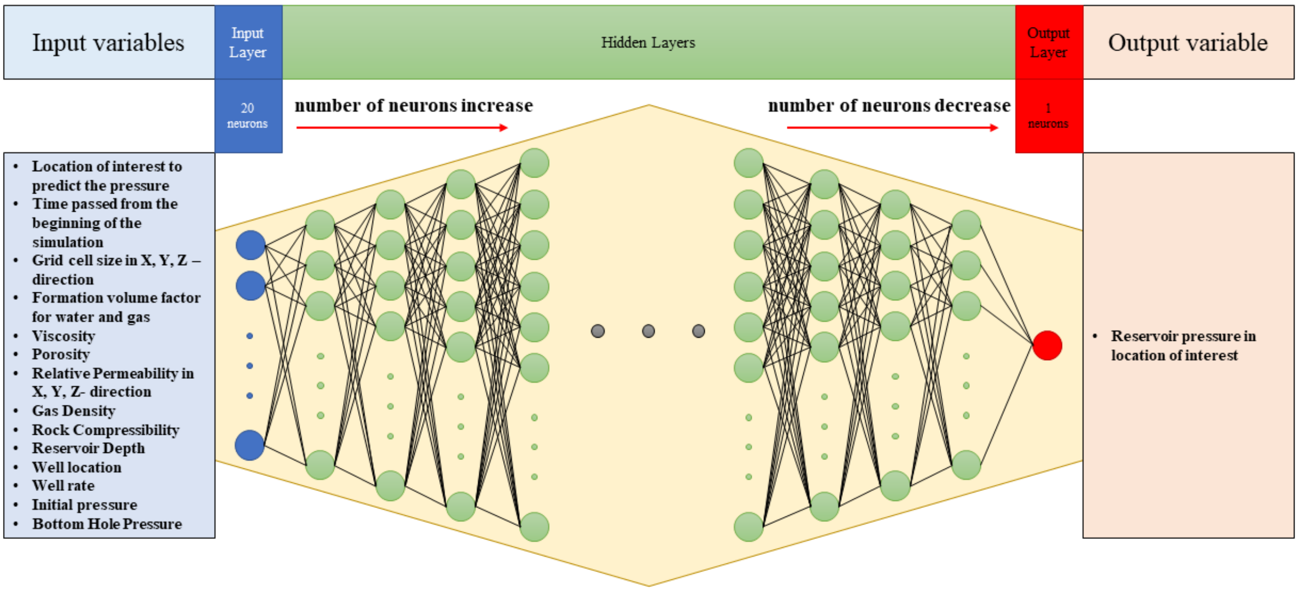

Input variables
Input Layer
Hidden Layers
Output Layer
Output variable
20 neurons
number of neurons increase
number of neurons decrease
1 neurons
• Location of interest to predict the pressure
• Time passed from the beginning of the simulation
• Grid cell size in X, Y, Z – direction
• Formation volume factor for water and gas
• Viscosity
• Porosity
• Relative Permeability in X, Y, Z- direction
• Gas Density
• Rock Compressibility
• Reservoir Depth
• Well location
• Well rate
• Initial pressure
• Bottom Hole Pressure
• Reservoir pressure in location of interest

Figure 2, Schematic diagram of the deep learning topology of DNS with dense layers

3. Results

In the process of developing the final model, many architectures (up to 25 layers) and parameters were attempted. Figure 3 represents the accuracy of the models with a different number of hidden layers compared with ECLIPSE. According to this figure, the most accurate model that we created is a model with 15 hidden layers with a mean absolute percentage error (MAPE) of 0.087 for the training dataset. It is worth noting that models designed with more than 15 layers did not provide more accurate results.

Although there were many architectures developed, to investigate the results further, the accuracy of four of these developed architectures was compared. Table 3 shows the attributes of these models. Model 3 represents the assumed final architecture. As explained within the methodology, these models demonstrate the highest level of accuracy that can be achieved with the corresponding number of hidden layers.

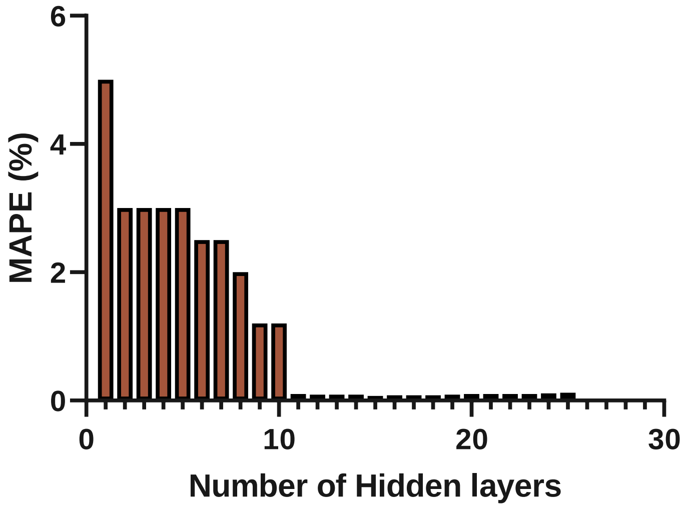


Figure 3, The effect of a different number of layers on the accuracy of the training dataset

Table 3, Comparing the structure of the three different models

Parameters	Model 1	Model 2	Model 3	Model 4
Layers	5	10	15	20
Neurons	2 - 100	2-100	2-300	2-300
Epoch	5000	5000	5000	5000
Regularisation	Drop out	Drop out	Drop out	Drop out
Activation functions	Linear	Linear, Elu and Relu	Elu and linear	Elu
Optimization Function	SGD	ADAM	ADAgrad	ADAgrad
Loss Function	MSE	MAPE	MSE	MSE

Throughout this article, in addition to MAPE and R-squared (R2), we used the mean relative error (MRE), mean square error (MSE) and root mean square error (RMSE) to evaluate the developed models. Figure 4 shows the MAPE percentage of the four aforementioned models during the learning process. Table 4 provides a comparison of MAPE, MAE, and MRE, MSE, RMSE, and R2 of the four models when evaluated with the test set.

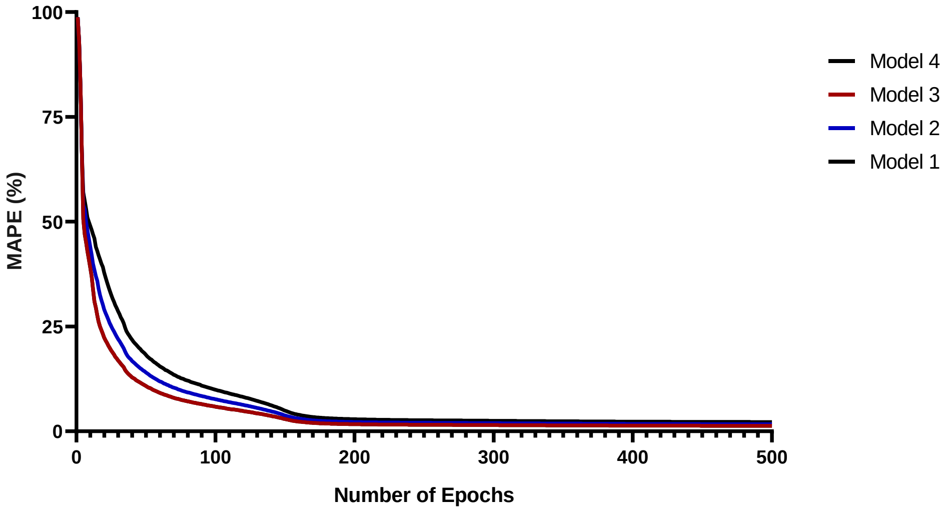


Figure 4, The change of accuracy (MAPE) of the models during training with 500 epochs

Table 4, Comparing the accuracy of the three different developed models using different metrics

Metrics	Model 1	Model 2	Model 3	Model 4
MAPE	3.011	1.210	0.891	0.890
MAE	121.254	38.179	15.906	15.201
MRE	0.015	0.010	0.006	0.006
MSE	42678.967	6732.207	3632.760	3629. 604
RMSE	206.589	82.050	60.272	60.246
R2	0.810	0.989	0.998	0.998

Given the accuracy of the models in Table 4, we were assured that the architecture used in model 3 is the best possible architecture for the database we used. As can be seen in Table 4, model 1 and model 2 were unable to reproduce the same values as the commercial simulator. In contrast, model 4 provided almost identical results with respect to model 3. However, the speed of model 3 is slightly faster when compared with model 4, as there are a lower number of hidden layers present.

Although the high value of R2 and low value of MAE and MAPE respectively are a promising sign of the accuracy of the model 3, the value of MSE is higher than we expected. As can be seen from the cross-validation plot, model 3 was able to predict the pressure values below 3000 psi and above 4000 psi with remarkable accuracy; however, it was unsuccessful in doing so for values between 3000 to 4000 psi. By studying the database, we realised this inaccuracy was caused by the lack of data points over these ranges. To solve this failure, we improved the dataset and included more data points for this specific margin. We added 20,000,000 more points to the data set in 4 stages. Figures 5a (top left) to 5d (bottom right) show the change of cross-validation plots through these four stages.

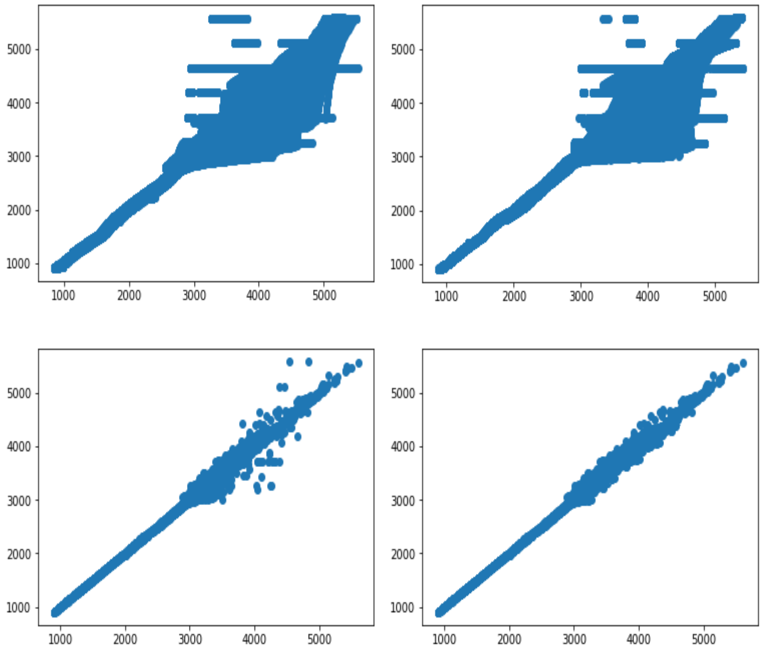


Figure 5, The change of the trend cross-validation plot when more data is used for training the model

By adding these data points, the quality of the databases improved, and features between variables were extracted more effectively by the deep learning model. After training the model with the new database, similar accuracies were achieved for the training, validation, and test sets: MAPE = 0.9, MAE = 11.0, MRE = 0.0, MSE = 651.3, RMSE =25.5, R2 = 1. Given the massive mismatch between model 3 and the final model, it is obvious that MSE and RMSE values must be considered as the main metrics of comparison in this problem and that low values of MAPE and MAE can be misleading. The reason the R2 values are high in all models is due to the number of data points we used. In other words, a high volume of accurate data points compensates for the less accurate values. The importance of this fact can be understood when comparing Figures 5a and 5d.

3.1.Benchmarking

We studied the effects of changing different input variables on the accuracy of DNS. Each benchmark includes a full simulation of a complete reservoir model. The initial reservoir pressure, porosity and permeability, well rate, and BHP were studied. Figure 6 compares the pressure calculated with ECLIPSE, with the pressure predicted using the developed model for the base model from Tables 5 and 6. As can be seen, the model accurately reproduced the pressure drop through different time steps for the base-model (MAE = 0.611, MAPE = 0.031, MRE = 0.000, MSE = 0.664, RMSE = 0.815, R2 = 1.000).

Table 5, Reservoir description for the base-model

Parameters (Oil Field Units) for Base-Model	**Values**
X, Y, Z-Direction Cell Size (ft)	100
Reservoir Depth (ft)	5000
Gas Density (lb/ft^3)	0.044
Initial Water Saturation (%)	0.20
Rock Compressibility @ 3500 psi (1/psi)	3.0E-06
Porosity (%)	0.20
Well rate (MSCF/d)	120,000
Time step (days)	10
Production period (days)	1000
Number of Cells in X and Y direction	50
Initial pressure (psia)	3000
Bottom Hole Pressure (psia)	1200

Table 6, PVT table for the base-model

Pressure	**Formation Volume Factor of gas (rb/MSCF)**	**Gas Viscosity (cP)**
1414	13.947	0.0124
1614	7.028	0.0125
1814	4.657	0.0128
2214	3.456	0.0130
2614	2.240	0.0139
3014	1.638	0.0148
3614	1.282	0.0161

To investigate the accuracy of the DNS, we designed different reservoir configurations based on the base-model. In these scenarios, different initial conditions, geology conditions and production settings were used to generate unseen reservoir models to examine the DNS. This model can be used to predict the pressure of a reservoir with unseen configurations without requiring further training and rebuilding, as long as the reservoir configuration lies within the range of the training data, Table 1. The primary benefit of such an approach is when a considerable number of simulations (i.e., in the range of thousands) are demanded in a limited time or with limited computing resources.

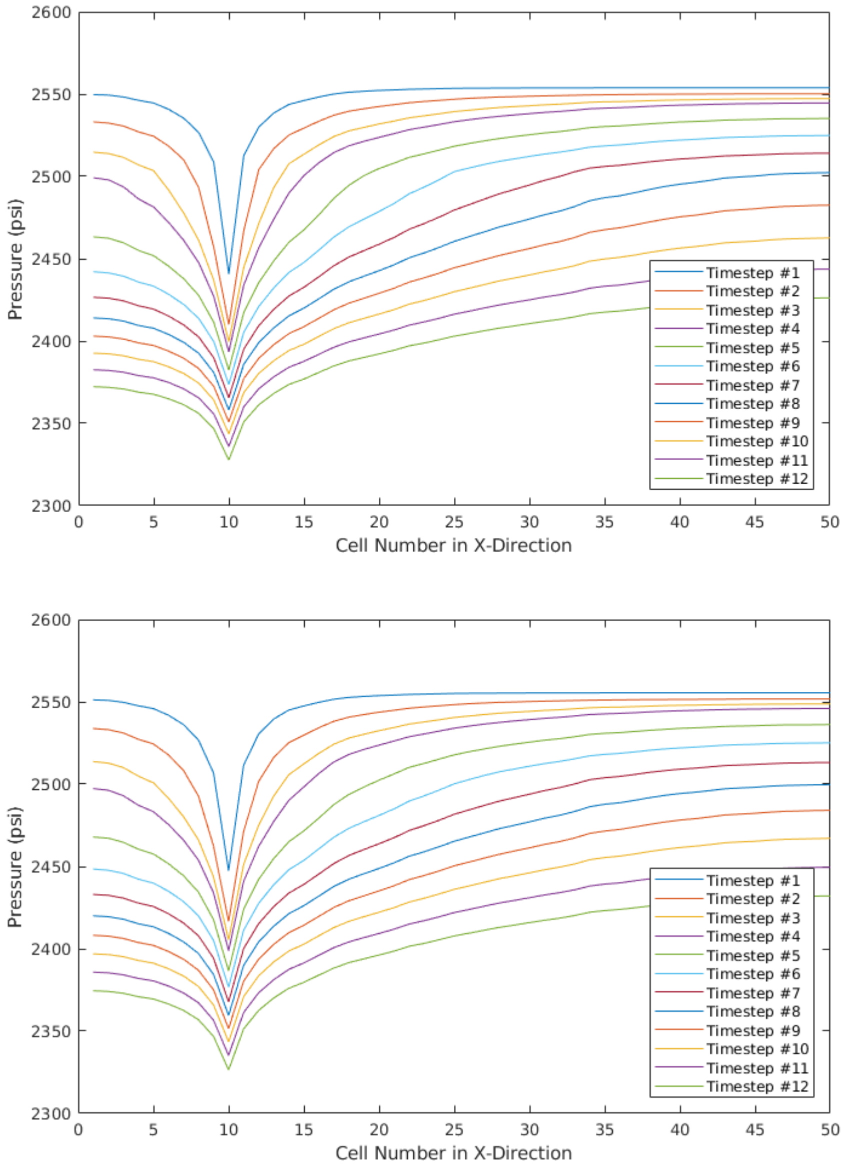


Figure 6, Pressure distribution throughout the reservoir for different time steps using DNS (up) and ECLIPSE (down)

3.1.1. Initial conditions:

Pressure:

Table 7 shows the accuracy of the model when we keep all the variables constant except for the initial pressure. As can be seen, for initial reservoir pressure less than 4000 psi, the model was incredibly accurate, but when we simulate reservoirs with an initial pressure higher than 4000 psi, it started to behave inconsistently. This issue relates to the fact that these values are beyond the range of the initial conditions we used in the training dataset. In other words, DNS is unsuccessful in extrapolating beyond the range of the training data set, which is not unusual in data-driven models. Due to the reliable results obtained via the current version of DNS for a given training dataset range, we believe that, given a broader range of training sets and a higher degree of computation power, an updated version of DNS can outperform the current version.

It must be mentioned that due to our own limited computational power, we could not surpass using 40,000,000 data points in the current study.

Table 7, The effects of different initial pressures on the accuracy of DNS

Initial pressure (psia)	**MAE**	**MAPE**	**MRE**	**MSE**	**RMSE**	**R2**
950	16.183	0.931	-0.003	438.778	20.947	0.998
1425	6.775	0.497	0.005	47.867	6.919	0.996
1900	14.296	0.913	0.009	259.473	16.108	0.998
2375 (Base Model)	0.224	0.013	0.000	0.260	0.510	1.000
2850	23.394	0.825	0.008	4590.823	67.756	1.000
3325	35.022	1.116	0.010	7678.560	87.627	0.991
3800	130.408	3.704	0.016	50274.049	224.219	0.755
4275	380.949	10.501	0.008	178086.970	422.004	0.157
4750	488.274	10.787	0.101	295087.841	543.220	0.415
5225	833.426	25.938	-0.259	910401.347	954.150	0.099

3.1.2. **Reservoir rock properties**:

Porosity and permeability:

The distribution of porosity and permeability can be highly variable across a reservoir. This variation is dependent on the geological settings of the reservoir. For example, in a sandstone reservoir, porosity is evenly distributed, with the permeability relatively high across the reservoir. However, in a carbonate reservoir, porosity and permeability depend on vugs and fractures, which are not uniformly distributed across the reservoir volume. Due to limitations in measuring these variables, there is a high level of uncertainty in the recorded values. A reliable simulator must provide an appropriate response to these uncertainties, as a minor change in the value could greatly influence the outcome. Therefore, we studied the effect of different distributions of porosity and permeability on the accuracy of DNS.

In Figure 7, there are four different porosity and permeability distributions present. In the top left and top right figures of Figure 7, porosity images and permeability reduce towards the right and bottom sides. The bottom left figure is of a constant value, while the bottom right figure is of a random distribution. In the latest scenario, the specific mean porosity for each case was considered, and the normal distribution function was applied to assign porosity over the grids in the reservoir. Then, based on the assigned value for porosity, a value for permeability was assigned to each cell using Timur’s equation (eq 1).

Table 8 shows the accuracy of DNS for the first three geological scenarios shown in Figure 7. Table 9 presents how a change of porosity affects the accuracy of DNS in more challenging cases when porosity is randomly distributed across the reservoir. As can be seen, DNS predicts consistent results for different porosity and permeability distributions. As shown in Tables 8 and 9, in all cases, the DNS predictions are highly accurate, unlike the results from ECLIPSE.

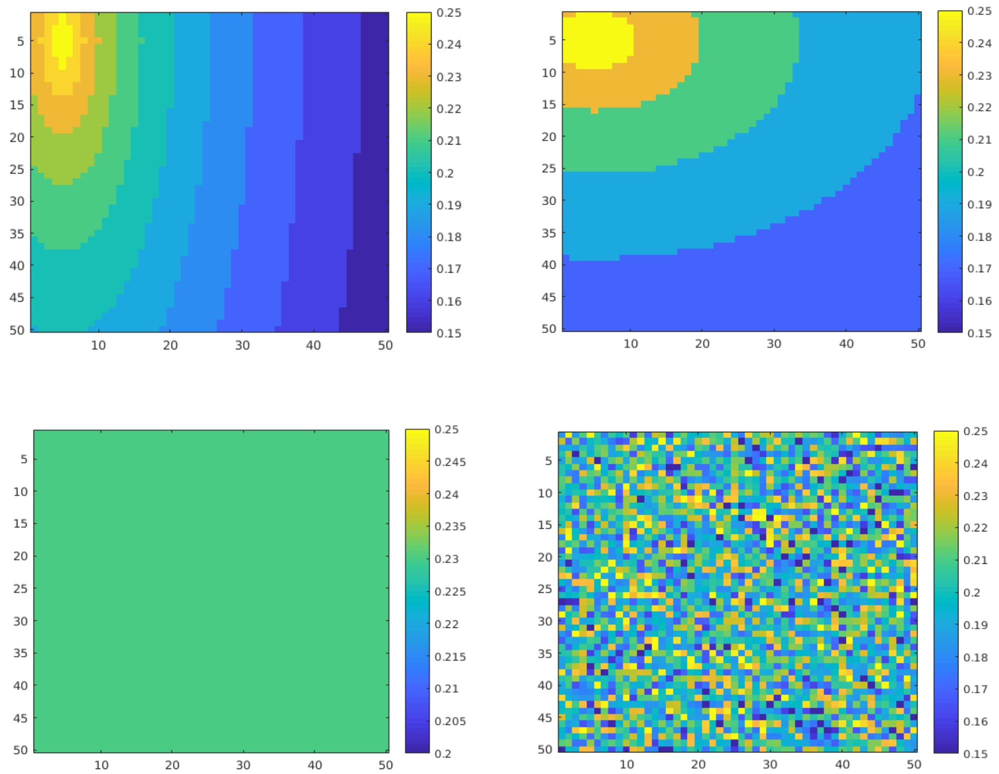


Figure 7, An illustration of the different porosity distributions we used in this study

Table 8, The effects of different porosity and permeability on the accuracy of DNS

Porosity	MAE	MAPE	MRE	MSE	RMSE	R2
Geology 1	4.407	0.248	-0.002	20.551	4.533	1.000
Geology 2	4.407	0.248	-0.002	20.551	4.533	1.000
Geology 3	4.729	0.260	-0.003	23.184	4.815	1.000

Table 9, The effects of different porosity and permeability on the accuracy of DNS in heterogeneous formations

Porosity	MAE	MAPE	MRE	MSE	RMSE	R2
0.15	0.444	0.025	0.000	0.636	0.797	1.000
0.16	0.261	0.015	0.000	0.364	0.604	1.000
0.17	0.287	0.016	0.000	0.351	0.593	1.000
0.18	0.261	0.015	0.000	0.360	0.600	1.000
0.19	0.244	0.014	0.000	0.275	0.525	1.000
0.2	0.232	0.013	0.000	0.298	0.546	1.000
0.21	0.197	0.011	0.000	0.213	0.462	1.000
0.22	0.339	0.019	0.000	0.456	0.676	1.000
0.23	0.222	0.013	0.000	0.229	0.479	1.000
0.24	0.228	0.013	0.000	0.296	0.544	1.000

3.1.3. Production Settings:

Well location

Wellbores were placed in various locations to evaluate the response given by DNS when the distance between the wellbore and boundaries varies. Table 10 demonstrates the accuracy of DNS compared with ECLIPSE when the different well locations are selected across the base model.

Table 10, the effects of different well locations on the accuracy of DNS

Well location (x,y)	MAE	MAPE	MRE	MSE	RMSE	R2
5,5 (Base Model)	0.224	0.013	0.000	0.260	0.510	1.000
5,10	16.892	0.938	0.009	517.064	22.739	1.000
10,10	13.513	0.750	0.007	413.651	20.338	1.000
10,15	14.639	0.813	0.008	448.122	21.169	1.000
15,15	16.892	0.938	0.009	517.064	22.739	1.000
20,20	18.018	1.000	0.010	551.534	23.485	1.000
25,20	20.270	1.125	0.011	620.476	24.909	1.000
30,15	21.396	1.188	0.011	654.947	25.592	1.000
35,10	13.513	0.750	0.007	413.651	20.338	1.000
35,35	15.765	0.875	0.008	482.593	21.968	1.000

Well rate

Table 11 represents the accuracy of DNS compared with ECLIPSE for different well rates. It is evident that, as in the previous case, changing the production rate does not affect the accuracy of DNS.

Table 11, the effects of different well rates on the accuracy of DNS

Well rate(Mscf/d)	MAE	MAPE	MRE	MSE	RMSE	R2
25000	0.522	0.028	0.000	0.664	0.815	1.000
50000	0.340	0.019	0.000	0.452	0.672	1.000
75000	0.264	0.015	0.000	0.290	0.539	1.000
100000	0.225	0.013	0.000	0.237	0.486	1.000
125000	0.225	0.013	0.000	0.272	0.522	1.000
150000	0.212	0.012	0.000	0.259	0.509	1.000
200000	0.210	0.012	0.000	0.279	0.528	1.000
250000	0.214	0.012	0.000	0.309	0.556	1.000
300000	0.233	0.013	0.000	0.362	0.602	1.000
400000	0.254	0.015	0.000	0.437	0.661	1.000

Bottom hole pressure (BHP)

Table 12 shows the effect of BHP on the accuracy of DNS. According to these results, changing the BHP does not affect accuracy. DNS has understood the influence of the determined BHP and predicts the correct value accordingly.

Table 12, the effects of different BHP on the accuracy of DNS

BHP (PSI)	**MAE**	**MAPE**	**MRE**	**MSE**	**RMSE**	**R2**
800	0.206	0.012	0.000	0.242	0.492	1.000
850	0.208	0.012	0.000	0.247	0.497	1.000
950	0.206	0.012	0.000	0.243	0.493	1.000
1000	0.208	0.012	0.000	0.249	0.499	1.000
1050	0.208	0.012	0.000	0.247	0.497	1.000
1100	0.213	0.012	0.000	0.254	0.504	1.000
1150	0.215	0.012	0.000	0.254	0.504	1.000
1200	0.224	0.013	0.000	0.260	0.510	1.000
1250	0.225	0.013	0.000	0.261	0.511	1.000
1300	0.240	0.014	0.000	0.269	0.519	1.000

Despite the accuracy that DNS has shown in these benchmarks, we are aware that our research may have some limitations: the first being that we only consider single-phase flow. The second involved the use of only one vertical production well that is considered within the reservoir. The third is that we did not attempt to simulate different geological structures. Regardless of these limitations, we believe that the results are promising and that by increasing the range of training data and improving computational power, we can eventually build a pre-trained simulator that can predict the fluid behaviour in any reservoir around the world, similar to any commercial numerical simulators.

4. Conclusion

This paper describes a data-driven model that predicts the pressure across a drainage area of a vertical well in a dry gas reservoir. The novelty of this study lies in the implementation of deep learning to develop a trained model with the capability to predict reservoir behaviour for a wide range of reservoir configurations. The results confirm that this approach is a promising method for building a pre-trained data-driven simulator with accurate, fast and reliable results for any reservoir configuration.

Unlike the conventional approach to proxy modelling, DNS is not built for one specific reservoir alone. Instead, once it is built, it can be used to predict the reservoir performance for a wide range of reservoir configurations, without further training and building processes.

The developed model showed similar results for the training, validation and tests sets, MAPE = 0.889, MAE = 11.014, MRE = 0.005, MSE = 651.276, RMSE =25.520, R2 = 0.999. We compared the results using different metrics to measure accuracy, as partial use of some of these metrics could be misleading.

A series of blind case studies in a wide range of different descriptions and settings are implemented to highlight the performance of the developed simulator, a deep net simulator. DNS predicts the pressure across the drainage area with remarkable accuracy when the model configurations lie within the training data used in this study (in the least accurate case: MAE<130 psi, MAPE<3.7%, MRE<0.016 psi, MSE<50275, RMSE<225 R2>0.76).

Prior to this study, machine learning has been used to build proxy models that represent only one reservoir. This study shows that with recent advances in machine learning algorithms, it is possible to create a proxy model not just for one reservoir but also for multiple reservoirs. Currently, DNS works only under the assumption of having one vertical wellbore in a single-phase flow reservoir. Despite these limitations, this ongoing study aims to develop a data-driven model that can act as a proxy model for a numerical reservoir simulator. This fast simulator can be used to investigate many cases during history matching, uncertainty quantification and field optimisation, leading to a more effective and lower cost decision-making process.

Acknowledgements

The authors are grateful to Alexander Arthurson for his assistance.

References:

Alenezi, F. and Mohaghegh, S., 2016. A data-driven smart proxy model for a comprehensive reservoir simulation, 2016 4th Saudi International Conference on Information Technology (Big Data Analysis)(KACSTIT). IEEE, pp. 1-6.

Alenezi, F. and Mohaghegh, S., 2017. Developing a Smart Proxy for the SACROC Water-Flooding Numerical Reservoir Simulation Model, SPE Western Regional Meeting. Society of Petroleum Engineers, Bakersfield, California.

Antonio Cardoso, M. and J. Durlofsky, L., 2010. Use of Reduced-Order Modeling Procedures for Production Optimisation, 15, 426-435 pp.

Artun, E., 2017. Characterising interwell connectivity in waterflooded reservoirs using data-driven and reduced-physics models: a comparative study. Neural Computing and Applications, 28(7): 1729-1743.

Artun, E., Ertekin, T., Watson, R. and Al-Wadhahi, M., 2011. Development of universal proxy models for screening and optimisation of cyclic pressure pulsing in naturally fractured reservoirs. Journal of Natural Gas Science and Engineering, 3(6): 667-686.

Baker, R.O., Yarranton, H.W. and Jensen, J., 2015. Practical reservoir engineering and characterisation. Gulf Professional Publishing.

Beckner, B.L., Haugen, K.B., Maliassov, S., Dyadechko, V. and Wiegand, K.D., 2015. General Parallel Reservoir Simulation, Abu Dhabi International Petroleum Exhibition and Conference. Society of Petroleum Engineers, Abu Dhabi, UAE.

Brown, S.P., Krupnick, A. and Walls, M.A., 2009. Natural gas: a bridge to a low-carbon future. Issue brief: 09-11.

Chen, B., He, J., Wen, X.-H., Chen, W. and Reynolds, A.C., 2017. Uncertainty quantification and value of information assessment using proxies and Markov chain Monte Carlo method for a pilot project. Journal of Petroleum Science and Engineering, 157: 328-339.

Eldred, M.E. et al., 2014. Reservoir Simulations in a High Performance Cloud Computing Environment, SPE Intelligent Energy Conference & Exhibition. Society of Petroleum Engineers, Utrecht, The Netherlands.

Ghasemi, M., Yang, Y., Gildin, E., Efendiev, Y. and Calo, V., 2015. Fast Multiscale Reservoir Simulations using POD-DEIM Model Reduction, SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, Houston, Texas, USA, pp. 18.

Ghassemzadeh, S. and Charkhi, A.H., 2016. Optimisation of integrated production system using advanced proxy based models: A new approach. Journal of Natural Gas Science and Engineering, 35: 89-96.

Ghassemzadeh, S., Perdomo, M.G. and Haghigh, M., 2019. Application of Deep Learning in Reservoir Simulation, Petroleum Geostatistics 2019.

Goodwin, N., 2015. Bridging the Gap Between Deterministic and Probabilistic Uncertainty Quantification Using Advanced Proxy Based Methods, SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, Houston, Texas, USA.

Guo, B. and Ghalambor, A., 2014. Natural gas engineering handbook. Elsevier.

Hagoort, J., 1988. Fundamentals of gas reservoir engineering, 23. Elsevier.

He, J., Xie, J., Wen, X.-H. and Chen, W., 2016. An alternative proxy for history matching using proxy-for-data approach and reduced order modeling. Journal of Petroleum Science and Engineering, 146: 392-399.

Iwere, F.O., Moreno, J.E. and Apaydin, O.G., 2006. Numerical Simulation of Thick, Tight Fluvial Sands. SPE-90630-PA, 9(04): 374-381.

Kalantari-Dahaghi, A., Mohaghegh, S. and Esmaili, S., 2015. Data-driven proxy at hydraulic fracture cluster level: A technique for efficient CO2- enhanced gas recovery and storage assessment in shale reservoir. Journal of Natural Gas Science and Engineering, 27: 515-530.

Kaleta, M.P., Hanea, R.G., Heemink, A.W. and Jansen, J.-D., 2011. Model-reduced gradient-based history matching. Computational Geosciences, 15(1): 135-153.

Kerr, R.A., 2010. Natural Gas From Shale Bursts Onto the Scene. Science, 328(5986): 1624.

Kim, M. and Shin, H., 2018. Development and application of proxy models for predicting the shale barrier size using reservoir parameters and SAGD production data. Journal of Petroleum Science and Engineering, 170: 331-344.

Kulga, B., Artun, E. and Ertekin, T., 2017. Development of a data-driven forecasting tool for hydraulically fractured, horizontal wells in tight-gas sands. Computers & Geosciences, 103: 99-110.

Kulga, B., Artun, E. and Ertekin, T., 2018. Characterisation of tight-gas sand reservoirs from horizontal-well performance data using an inverse neural network. Journal of Natural Gas Science and Engineering, 59: 35-46.

Leung, G.C.K., 2015. Natural Gas as a Clean Fuel, Handbook of Clean Energy Systems, pp. 1-15.

Mohaghegh, S.D., Gaskari, R. and Maysami, M., 2017. Shale Analytics: Making Production and Operational Decisions Based on Facts: A Case Study in Marcellus Shale, SPE Hydraulic Fracturing Technology Conference and Exhibition. Society of Petroleum Engineers, The Woodlands, Texas, USA.

Nwachukwu, A., Jeong, H., Pyrcz, M. and Lake, L.W., 2018. Fast evaluation of well placements in heterogeneous reservoir models using machine learning. Journal of Petroleum Science and Engineering, 163: 463-475.

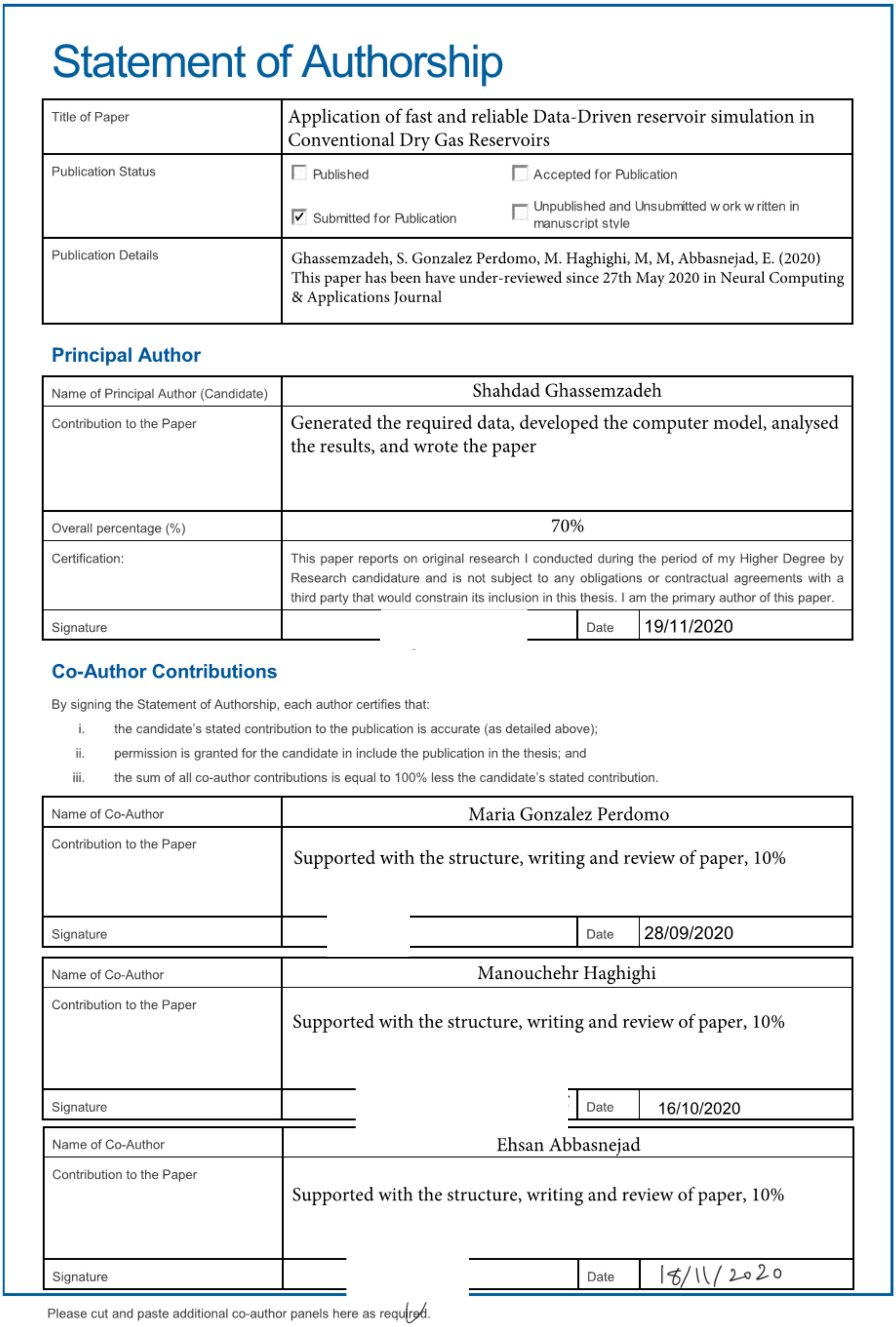
Schön, J.H., 2015. Physical properties of rocks: Fundamentals and principles of petrophysics. Elsevier.

Timur, A., 1968. An investigation of permeability, porosity, & residual water saturation relationships for sandstone reservoirs. The Log Analyst, 9(04).

Tour, J.M., Kittrell, C. and Colvin, V.L., 2010. Green carbon as a bridge to renewable energy. Nature materials, 9(11): 871-874.

Zou, C. et al., 2012. Tight gas sandstone reservoirs in China: characteristics and recognition criteria. Journal of Petroleum Science and Engineering, 88-89: 82-91.

5. Application of fast and reliable Data-Driven reservoir simulation in Conventional Dry Gas Reservoirs



Statement of Authorship

Title of Paper	Application of fast and reliable Data-Driven reservoir simulation in Conventional Dry Gas Reservoirs
Publication Status	□ Published □ Accepted for Publication ☑ Submitted for Publication □ Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	Ghassemzadeh, S. Gonzalez Perdomo, M. Haghighi, M, M, Abbasnejad, E. (2020) This paper has been have under-reviewed since 27th May 2020 in Neural Computing & Applications Journal

Principal Author

Name of Principal Author (Candidate)	Shahdad Ghassemzadeh		
Contribution to the Paper	Generated the required data, developed the computer model, analysed the results, and wrote the paper		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	19/11/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Maria Gonzalez Perdomo		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	28/09/2020

Name of Co-Author	Manouchehr Haghighi		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	16/10/2020

Name of Co-Author	Ehsan Abbasnejad		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	18/11/2020

Please cut and paste additional co-author panels here as required.

Abstract

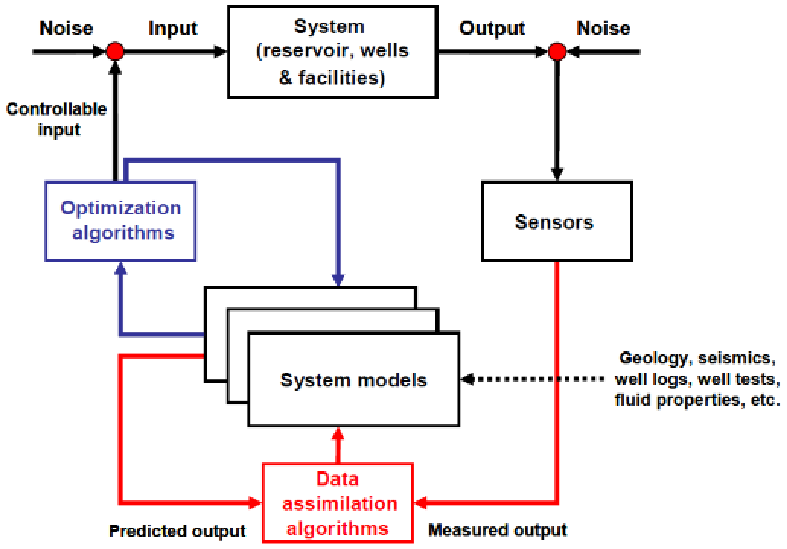
Physics-based reservoir simulation is the backbone of many decision-making processes in the oil and gas industry. However, due to being computationally demanding, simulating a model multiple times in iterative studies, such as history matching and production optimisation, is extremely time-intensive. This downside results in it being practically impossible to update the model every time a set of new data is available. One of the popular solutions for this problem is creating an approximate proxy model of the desired reservoir that combines or replaces the physics-based model with this fast alternative. However, the consequence of this approach is that such a proxy model can only represent one corresponding reservoir, and, for every new reservoir, a new proxy model must be rebuilt from scratch. Additionally, when the overall runtime is considered, it can be observed that, in some cases, iteratively running a numerical reservoir simulation may be quicker than the process of building, validating, and using a proxy model. To overcome this obstacle, in this study, we used deep learning to create a data-driven simulator, deep net simulator (DNS), that enables us to simulate a wide range of reservoirs. Unlike the conventional proxy approach, we collected the training data from a large set of observations, which consist of multiple reservoirs with completely different configurations and settings. The hypothesis is that such an approach can teach the DNS to learn the principles of modelling a reservoir and act as an excellent approximator to the equations that a physics-based numerical simulator solves. We compared the precision and reliability of DNS with a commercial simulator for 600 generated case studies, consisting of 500,000,000 data points. DNS successfully predicts 45%, 70% and 90% of the cases with a mean absolute percentage error of less than 5%, 10% and 15% respectively. Due to the indirect dependency of DNS on the initial and boundary conditions, DNS can be executed for a specific cell for a specified period. This attribute allows DNS to act incredibly fast when compared with traditional physics-based simulators. Our results showed that DNS is, on average, 9.25E+7 times faster than a commercial simulator. Furthermore, if one simulates the entire grid cells with both DNS and ECLIPSE, DNS would still be relatively faster (37% on average). With the same consideration of simulating the entire reservoir, when compared with ECLIPSE, DNS has shown that when two relatively large-scaled case studies were simulated 1000 times, DNS could reduce the runtimes by 108% and 443% (56 and 177 hours), respectively.

1. Introduction

Reservoir modelling is a crucial aspect of any decision-making aspect within reservoir management studies and field development planning. Generally, when a reservoir model is being built, one of the following methods are used: analogical, experimental or physics-based numerical methods (Ertekin, Abou-Kassem et al. 2001).

If the required data is available, physics-based numerical simulation is the most popular method to predict reservoir performance. Within the past two decades, the expansion of implementing automation within different operations and services has evolved significantly within the petroleum industry. This expansion has led to the development of smart field technologies that enable the almost continuous collection of various data. However, as a result of this large data income, a new challenge has arisen as to how best to use this big data in order to create a useful value and thus make better decisions. The closed-loop reservoir management approach is recommended as an efficient approach to overcome the challenges faced when a massive amount of data is present, Figure 1(Jansen, Brouwer et al. 2009). This approach involves updating models continuously as soon as new data is available. When using this workflow in decision-making, it is crucial that the model can fulfil the following tasks:

- Update quickly once new data is presented.
- Be accurate enough to represent the actual system.



Noise
Input
System (reservoir, wells & facilities)
Output
Noise
Controllable input
Optimization algorithms
Sensors
System models
Geology, seismics, well logs, well tests, fluid properties, etc.
Data assimilation algorithms
Predicted output
Measured output

Figure 1, A scheme of the closed-loop reservoir management approach (Jansen, Brouwer et al. 2009)

However, physics-based numerical simulation is not a great candidate for this scheme due to its intense computational nature. Because of this attribute, the simulation of a full reservoir takes a significant period of time, ranging from a couple of hours to days. This is due to the discretisation process required to model the reservoir and the iterative nature of the optimisations and complex partial differential equations that have to be solved within the simulator. Although this runtime can be partially reduced with the use of high-speed computers, there is still a considerable associated cost in doing this (Eldred, Orangi et al. 2014, Beckner, Haugen et al. 2015).

In order to address the required precision of a potential approach in this workflow, two elements must be considered: the accuracy of the method itself and the accuracy of the input data.

For the first element, reservoir management must be looked at from a business and economic perspective. In practice, this means, based on the aim defined within a particular study, individual decisions are made to determine the level of precision required to fulfil the defined objective. If less expensive techniques can provide satisfactory results, then more advanced and costly methods are not justified (Coats 1969, Fanchi 2001).

Additionally, regarding the accuracy of the input data, due to the presence of either limited data or measurement errors, both static and dynamic input data generate considerable levels of uncertainty. This level of uncertainty within the inputs leads to partially inconsistent outputs in comparison with field production data. Therefore, to better quantify the uncertainties present during performance predictions and thus better manage its associated risk, hundreds to thousands of simulations must be completed. Furthermore, a similar number of simulations is required to complete both the history matching and the field optimisation processes. The justification for this iterative approach is that each task attempts a variety of parameters and operation strategies. This number of simulations, combined with the amount of time taken to complete each respective simulation, results in an extremely time-consuming process, becoming a liability to engineers and companies alike.

To reduce this computational time and the cost associated with the simulations, many researchers suggest building a proxy model of the desired reservoir and use its fast approximation instead (Jensen 2017, Mohaghegh 2017, Abooali and Khamehchi 2019, Zheng, Leung et al. 2019). They present different approaches, mostly including machine learning methods. Through the application of simple mathematics functions, the proxy model of a corresponding reservoir can approximate the response of that system quickly, without requiring a significant amount of computational simulations (Artun, Ertekin et al. 2011, Goodwin 2015, Kalantari-Dahaghi, Mohaghegh et al. 2015, Alenezi and Mohaghegh 2016, Ghassemzadeh and Charkhi 2016, He, Xie et al. 2016, Alenezi and Mohaghegh 2017, Chen, He et al. 2017, Mohaghegh, Gaskari et al. 2017, Kim and Shin 2018, Nwachukwu, Jeong et al. 2018). Model inputs usually include initial conditions, operational parameters, and reservoir characteristics such as porosity, permeability, etc. Model outputs include the production or saturation profile, recovery factor, etc.

While these methods have provided a significant step forward in the field, each of these proxy models can only be used for their corresponding reservoir. This means that for each new reservoir, a new proxy model must be constructed from scratch. Furthermore, if the time spent building, validating, and running a proxy model is considered, it can be observed that running a numerical reservoir simulation iteratively could result in being comparatively more time efficient.

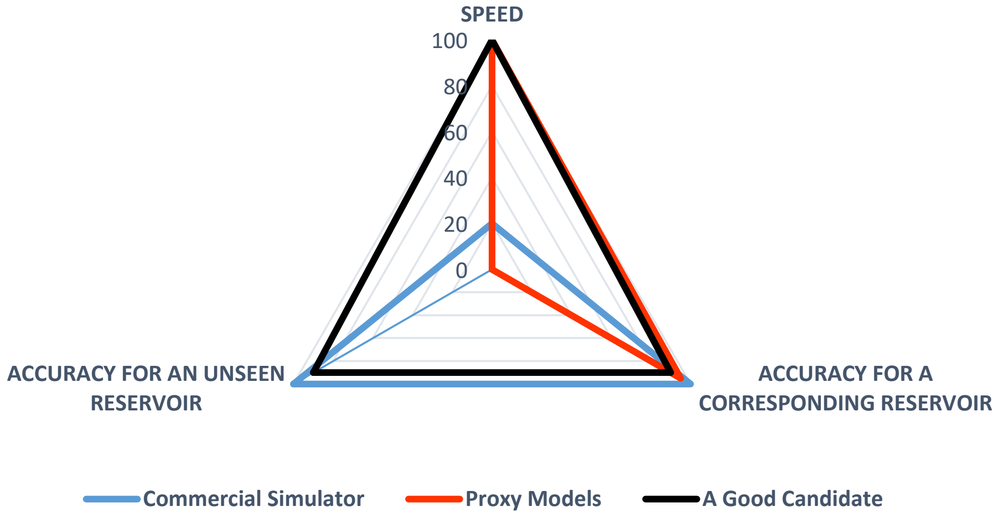


Figure 2, A rough comparison of the accuracy and speed of a commercial simulator, a proxy model and a possible suitable replacement.

A good candidate, which can satisfy both prerequisites of the closed-loop reservoir management approach, must have both the accuracy and generalisation of a commercial simulator and the speed of the proxy models, Figure 1.2. This article seeks to expand our previous work on the application of deep learning on reservoir simulation into a three-dimensional space (Ghassemzadeh, Perdomo et al. 2019). The aim for this model is for it to be used as a forecasting tool: to instantly predict the pressure at any point within the reservoir once given the initial conditions, operation parameters and reservoir characteristics. An advantage of using such a simulator is that it enables us to investigate a wide range of scenarios and thus present a more profitable production scheme. This approach is to be interpreted as a data-driven clone to the numerical reservoir simulation, representing multiple reservoirs as opposed to the single reservoir approach dictated by conventional proxy modelling. Therefore, the outcome of this research can be considered a data-driven simulator. This simulator is pre-trained and independent of a commercial simulator. Ideally, we should learn directly from observations, as this allows us to reduce the uncertainty associated with the fluid flow equations, but we use the simulators as the surrogate in the experiments here.

2. Methodology

2.1. Deep Net Simulator (DNS)

The steps that were taken in order to develop this proposed model are in some way similar to the steps previous researchers have proposed in their development of a conventional proxy model representing a specified reservoir. However, the database that was used in the training of our model was collected from a range of different reservoir models instead of only one reservoir model. This approach results in an elaborated database covering a range of complexities that may occur within a reservoir, Figure 3. This approach can be used to help us create more sophisticated, complex, data-driven models that can be used as proxies to various numerical reservoir simulations.

The main challenge throughout this study was to train such complicated problems and to extract the complex features from amongst this big data. We achieved this by relying on one of the latest machine learning techniques, deep learning.

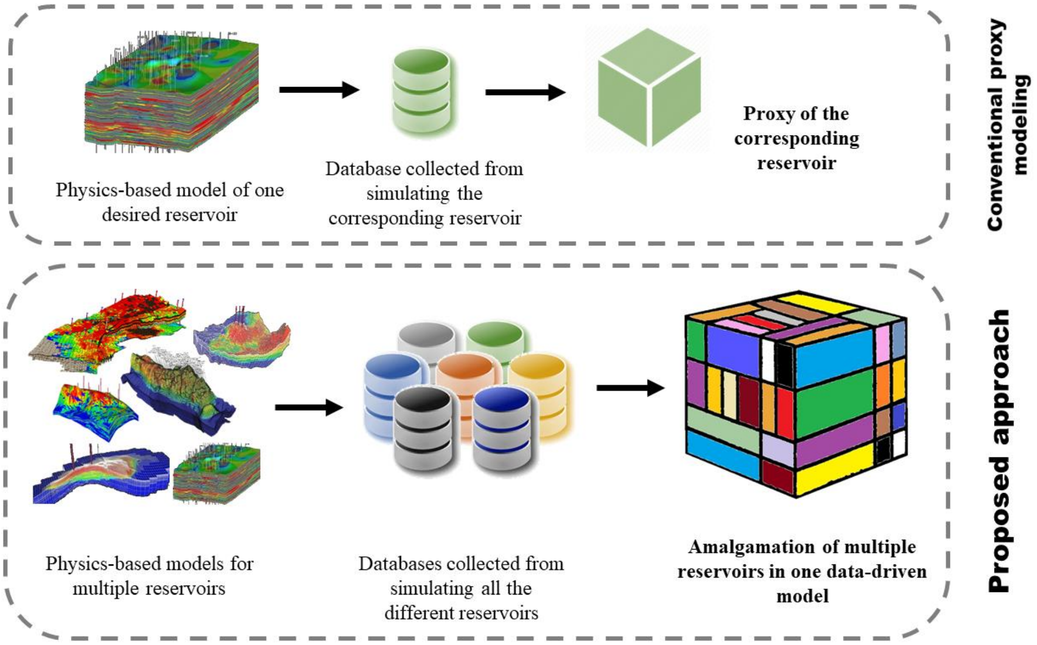

Physics-based model of one desired reservoir
Database collected from simulating the corresponding reservoir
Proxy of the corresponding reservoir
Conventional proxy modeling
Physics-based models for multiple reservoirs
Databases collected from simulating all the different reservoirs
Amalgamation of multiple reservoirs in one data-driven model
Proposed approach

Figure 3, Flowchart of building DNS. This Figure demonstrates the difference between a traditional proxy model and a DNS model. While a proxy model corresponds to a single desired reservoir, Our DNS can be used as a proxy for a wide variety of reservoirs.

2.2.The development of a Spatiotemporal database

During this step, we generated a spatial-temporal database in order to train a deep learning model and, consequently, to replicate the behaviour of a commercial simulator. Future work in this study involves the use of real field data, to fine-tune the potential of our developed model to match the reality of the reservoir behaviour. This will allow us to reduce the uncertainty present within diffusivity equations. As there are different physical phenomena used to govern different reservoirs, the first step in developing an alternative tool for reservoir simulation is to target a specific reservoir type.

During this study, we have concentrated on natural gas reservoirs with gas expansion as the primary drive mechanism. There are key reasons to focus on gas reservoirs. First, from a sustainability point of view, many consider natural gas as a clean bridge to a renewable future (Brown, Krupnick et al. 2009, Kerr 2010, Tour, Kittrell et al. 2010, Leung 2015). Given how much natural gas resources are prospective; we prioritise gas reservoirs over oil resources. Furthermore, in comparison with single-phase oil flow, single-phase gas flow is more complicated, and therefore more challenging to model using a data-driven model. There are two main differences between the single-phase flow of oil and gas. First, gas flow velocities, in general, are higher to the extent that inertial forces may become significant and can no longer be ignored. Therefore, Darcy's Law for flow in porous media should be replaced by the more complex Law of Forchheimer, which includes the effect of inertial forces. Second, the gas compressibility and gas viscosity are dependent on the pressure, and this results in a differential equation for gas flow that is substantially non-linear (Hagoort 1988).

Since single-phase gas flow is investigated in this research, we selected a black oil model, which is accurate enough to formulate and represent the behaviour of gas reservoirs (Iwere, Moreno et al. 2006). Furthermore, the model we considered to simulate the fluid flow in the dry gas reservoir is a multi-layer (3-dimensional) system with rectangular, uniform grid cells distribution. We assumed the drainage area is partially saturated with gas and the water-phase is considered immobile.

We used a commercial reservoir simulator (ECLIPSE) to develop the spatial-temporal data of the gas reservoirs. Using this simulator, we built a template black oil model that represented the drainage area of a vertical well in a dry gas reservoir. Based on a predetermined range of reservoir configurations, we then generated the required spatial-temporal database.

In the development of our model, we first began with the random generation of 100 synthetic reservoir configurations. These models were designed with a random generator that randomly chooses values that lie within the pre-specified range of data presented in Table 1. These values are based on real field data obtained from various reservoirs located around the globe (Zou, Zhu et al. 2012). By using randomised realistic values, we aimed to generate synthetic reservoir configurations with various properties that would allow us to determine and describe the physical behaviour of gas flow in porous media.

The first step in each iteration of this flowchart was to create an ECLIPSE data file. ECLIPSE data files comprise the necessary reservoir descriptions and settings to model a reservoir. In order to create the proper data file, the required values were read from the range presented in Table 1 randomly, and written into a file in the data file format. It should be noted that each value obtained from each row in Table 1 was chosen independently from each value in other rows in order to maximise the randomness.

In order to make this model more realistic, we specified that the geology in our models was heterogeneous in every case. This allowed us to consider the specified mean porosity for each reservoir by applying the normal distribution function, allowing us to assign porosity values over the grids in the reservoir. In order to calculate the absolute permeability for each cell, we used one of the most common modifications of the Wyllie-Rose empirical equation, Timur's equation (Timur 1968). This equation, displayed below, can provide a reasonable estimation of absolute permeability for sandstone formations (Schön 2015):

$$\mathrm{k} = \left[100 \times \frac{\varphi^{2.25}}{\mathrm{S_{w,irr}}}\right]^2 \qquad (1)$$

where the $S_{w,irr}$ is the irreducible water saturation and is considered to be 20 % for all cases (Baker, Yarranton et al. 2015). This was followed by a calculation of the reservoir's initial pressure. To do this, we used hydrostatic pressure based on the assigned reservoir depth (Hagoort 1988). This is given by:

$$\mathrm{P_i} = 0.475 \times \mathrm{Depth} \qquad (2)$$

Apart from the previously-mentioned variables, some features and settings were fixed throughout all the scenarios and models. In this study, we considered that the well production was controlled by the bottom-hole pressure (BHP). This is a practical setting in a gas field, as it helps us to avoid sand production and water coning, and satisfies the constraints in regards to production facilities and regulatory authorities (Guo and Ghalambor 2014). Therefore, the well production present in each model was produced at its assigned well rate until it reaches its allocated BHP. It should be noted that both the saturation and relative permeability values were kept constant; this is due to this study being focused on single-phase flow.

After generating the data files, the ECLIPSE software was used to simulate the data files. A parsing algorithm was then used to read and convert the required values from the specified data files, and their corresponding output files were converted into tabular-format files. These tabular-format files were then combined into a single file, resulting in a database of approximately 40,000,000 data points, corresponding to all 100 synthetic models.

The training data files included:

Inputs:

- Time
- Cell size in X, Y, Z – direction
- Formation volume factor for water and gas
- Viscosity
- Porosity
- Relative permeability in the X, Y- direction
- Gas density
- Initial water saturation
- Rock compressibility

- Reservoir depth
- Well location
- Well rate
- Initial pressure
- Bottom hole pressure

Output:

- Cell pressure

Table 1, Range of different variables used to generate data sets

Parameters (Oil Field Units)	**Min**	**Max**
X, Y, Z-Direction Cell Size (ft)	50	300
Reservoir Depth (ft)	3,000	11,000
Gas Density (lb/ft^3)	0.038	0.048
Initial Water Saturation (%)	20	30
Formation Volume Factor (rb/MSCF)	0.355	180.349
Rock Compressibility @ Ref Pressure (1/psi)	3.00E-06	4.50E-06
Gas Viscosity (cP)	0.009	0.180
Permeability (md)	30	800
Porosity (%)	4	32
Well rate (MSCF/d)	1,200	1,000,000
Time step (days)	5	30
Number of Cells in XY Plane	100	4900
Initial pressure (psi)	1,425	5,225
Bottom Hole Pressure (psi)	14.7	1,400

Table 2, Different PVT tables used in this study

PVT1		
Pressure (psi)	Bg (rb/Mscf)	µg (cp)
1414	13.947	0.0124
1614	7.028	0.0125
1814	4.657	0.0128
2214	3.456	0.013
2614	2.24	0.0139
3014	1.638	0.0148
3614	1.282	0.0161

PVT2		
Pressure (psi)	Bg (rb/Mscf)	µg (cp)
1214	13.947	0.0124
1414	7.028	0.0125
1614	4.657	0.0128
1814	3.456	0.013
2214	2.24	0.0139
2614	1.638	0.0148
3014	1.282	0.0161

PVT3		
Pressure (psi)	Bg (rb/Mscf)	µg (cP)
14.700	180.349	0.009
513.965	4.588	0.011
1013.230	2.028	0.013
1512.495	1.174	0.019
2011.760	0.793	0.029
2511.025	0.619	0.043
3010.290	0.535	0.057
3509.555	0.489	0.070
4008.820	0.459	0.082
4508.085	0.438	0.093
5007.350	0.423	0.103
5506.615	0.410	0.112
6005.880	0.400	0.121

6505.145	0.392	0.130
7004.410	0.385	0.138
7503.675	0.378	0.145
8002.940	0.373	0.153
8502.205	0.368	0.160
9001.470	0.363	0.166
9500.735	0.359	0.173
10000.000	0.355	0.180

Table 3, Range of parameters used in the Deep Learning Network

Parameters	Values
Layers	Up to 25
Neurons	Up to 1000
Epoch	5000
Regularisation	Drop out
Activation functions	Linear, Sigmoid, tanh, ELU, Relu
Optimization Function	ADAgrad, ADAdelta, ADAM, SGD
Loss Function	MSE, MAPE, MAE

2.3.The training of the model

The next step is the determination of suitable architecture and the selection of a proper training configuration for the above-mentioned architecture. The database that was generated in the previous step is then fed to this architecture, such that the trained model is able to predict the same results as a numerical simulator.

We used dense-layered (fully-connected) deep learning and attempted different architectures to create the best possible model. We started with an architecture that contains one hidden layer and used a variety of different combination of neurons, activation functions, optimisation functions and loss functions. Once we achieved the best possible parameters within the architecture with one hidden layer, we increased the number of hidden layers to two and attempted to optimise the new architecture by attempting new values to these hyper-parameters. We continued to increase the layers until we were assured that there was no chance of improving the accuracy of the model by increasing the layers. We used an early stopping strategy with a validation set using the mean absolute percentage error (MAPE) as the monitoring metric in the training process to avoid the overfitting problem. Table 3 shows the different values we attempted for each parameter and hyperparameter to create the proposed model, and Table 4 shows the values for the final model. Figure 54 illustrates the schematic diagram of the deep learning topology we used in this study.

Table 4, Parameters and hyper-parameters used in the deep learning network

Parameters	Values
Layers	14
Neurons	Up to 300
Epoch	5000
Regularisation	Drop out
Activation functions	Linear, ELU
Optimization Function	ADAgrad
Loss Function	MSE

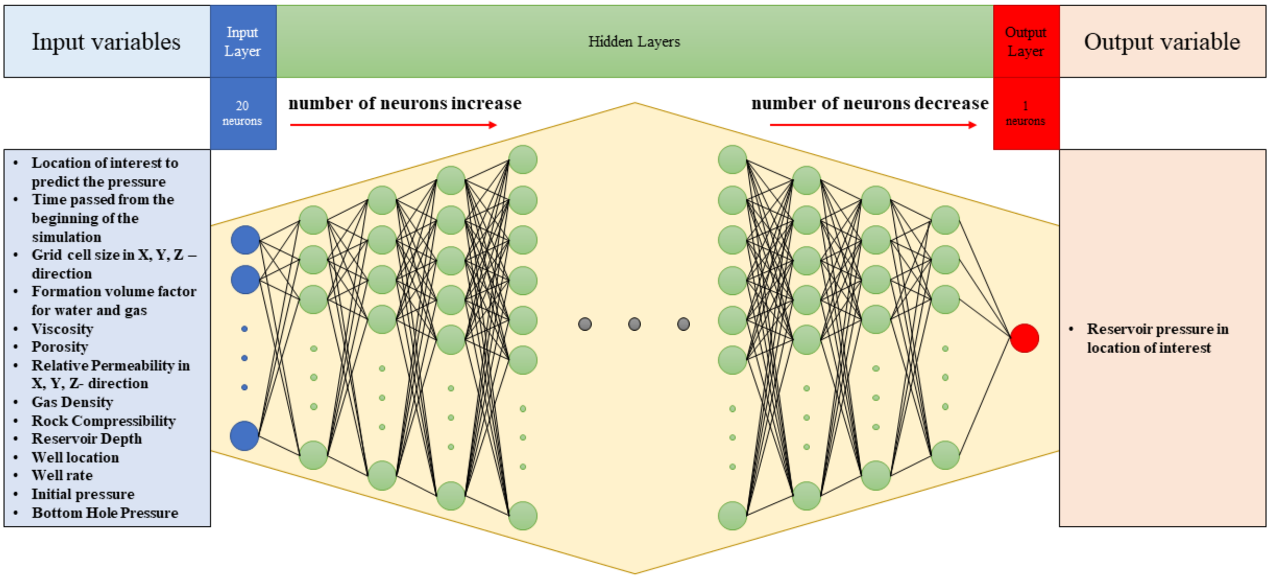

Input variables
Input Layer
Hidden Layers
Output Layer
Output variable
20 neurons
number of neurons increase
number of neurons decrease
1 neurons
Location of interest to predict the pressure
Time passed from the beginning of the simulation
Grid cell size in X, Y, Z – direction
Formation volume factor for water and gas
Viscosity
Porosity
Relative Permeability in X, Y, Z- direction
Gas Density
Rock Compressibility
Reservoir Depth
Well location
Well rate
Initial pressure
Bottom Hole Pressure
Reservoir pressure in location of interest

Figure 54, Schematic diagram of the deep learning topology of DNS with dense layers

2.4.Validation procedures

Data-driven models are usually validated using a test dataset. However, since we were aiming to create a stand-alone simulator, validating the model by using a test dataset alone was not deemed sufficient. To challenge the developed simulator, we created different case studies to examine the accuracy and speed of this developed simulator. These benchmarks help us to study the accuracy and speed of the simulator in each cell per production time and provide thoughtful feedback as to whether or not the developed simulator can both follow the trend of pressure drops and understand the physics of the problem correctly.

We used the following metrics to examine the accuracy performance of the developed simulator:

- Mean absolute error (MAE):

$$MAE = \frac{1}{n}\sum_{j=1}^{n} \left|y_j - \hat{y}_j\right| \tag{3}$$

- Mean absolute percentage error (MAPE)

$$MAPE = \frac{1}{n}\sum_{j=1}^{n} \left|\frac{y_j - \hat{y}_j}{y_j}\right| \tag{4}$$

- Mean Squared Error (MSE)

$$MSE = \frac{1}{N}\sum_{j=1}^{N}\left(y_j - \hat{\mathrm{y}}_{\mathrm{j}}\right)^2 \tag{5}$$

- Root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{\mathrm{y}}_{\mathrm{j}}\right)^2} \tag{6}$$

- R Squared (R2)

$$R2 = 1 - \frac{MSE\ (model)}{MSE\ (baseline)} \tag{7a}$$

The MSE of the model is computed as above, while the MSE of the baseline is defined as:

$$MSE\ (baseline) = \frac{1}{N}\sum_{j=1}^{N}\left(y_j - \bar{y}\right)^2 \tag{7b}$$

where $\hat{y}$ is the prediction, y is the observed value and $\bar{y}$ is the average of the observed values.

3. Results

This approach results in the creation of a deep learning model that has the speed of a proxy model and an acceptable generalisation when compared with a commercial simulator. In the following section, we present the results from a few of the case studies investigated using the developed simulator, Deep Net Simulator (DNS), and compare the results obtained from our developed simulator with a commercial simulator (ECLIPSE).

Overall, around 600 case studies, containing more than 500,000,000 data points have been used for testing the accuracy of DNS. By using these case studies, we were able to investigate the claims that we made earlier to develop an accurate, fast reservoir simulator. In these case studies, we began by investigating the accuracy of DNS. This resulted in DNS and ECLIPSE being used to simulate each of the reservoir models, calculating the pressure change for all cells over the production time. The speeds of the developed simulator and ECLIPSE are then compared for each case where the model contains a higher number of cells. Finally, we discussed the reliability of DNS.

- **Accuracy of the case studies:**

In each benchmark case, we covered a wide variety of possibilities. Cell sizes were randomised between 60 - 300 ft. The porosity among the grid cells was randomly distributed, with the mean porosity generated via a normal distribution function (Slider 1983). The values used for this mean porosity were directly related to the depth of the reservoir. The reservoir's initial pressure was calculated in proportion to the depth of the reservoir, with normal pressure regimes considered in each case (Hagoort 1988). However, as each benchmark case is a gas reservoir, the rock compressibility is not deemed an important parameter. The gas density in each case is considered between 0.038 to 0.042 for natural gas. As this study concentrates on dry gas, we can assume only one fluid is within the system. This means the initial water saturation is considered the minimum value.

Furthermore, we can assume the saturation table is constant for each benchmark case, as it has a low effect on the model. In some cases, a constant well rate was allocated over the production period. Conversely, in the rest, a high well rate was allocated such that, after 40% of the production period, the well rate would continuously drop for the duration of the reservoir's remaining life, satisfying the assigned bottom hole pressure (BHP). Values of the BHP were considered to lie between 900 to 1200 psi; these are typical values for conventional gas reservoirs. Each benchmark was simulated for up to five years, with time steps of up to 30 days. Different well locations were investigated in each case.

As five cells in the Z-direction were selected per case, different production scenarios were subsequently investigated. The grids that connected the respective wells were either fully or partially open, resulting in either all five cells being connected to the well or a randomly allocated amount being connected.

Due to each reservoir having varying grid sizes and porosities, this results in each reservoir having a different amount of gas originally in place. Furthermore, three PVT tables (Table 2) were randomly assigned to each case. Different values of formation volume factors and viscosity imply a different type of fluid flow in each reservoir.

- Case study 1:

Table 5 shows the properties of this case study. This model includes 4500 cells with an initial gas saturation of 80%. PVT1 in Table 2 shows the PVT properties of the gas. It is worth noting that there is no water flow within the porous media throughout its production time. The well produces for 600 days over 40 timesteps, meaning the pressure drop of each cell is reported every 15 days. We considered the gas reservoir is located in shallow depth and gas in-place has a density equal to 0.04 lb/ft3. A mean porosity of 0.15 is considered, and normal distribution was used to allocate the porosity to each cell within the 30 x 30 x 5 grid. Figure 5a and Figure 5b indicate the porosity and permeability (kx) distributions that were used to create this reservoir model. The wellbore is located in the corner of the grid with an open hole wellbore. The initial well rate was set to 120,000 Mscf/ day.

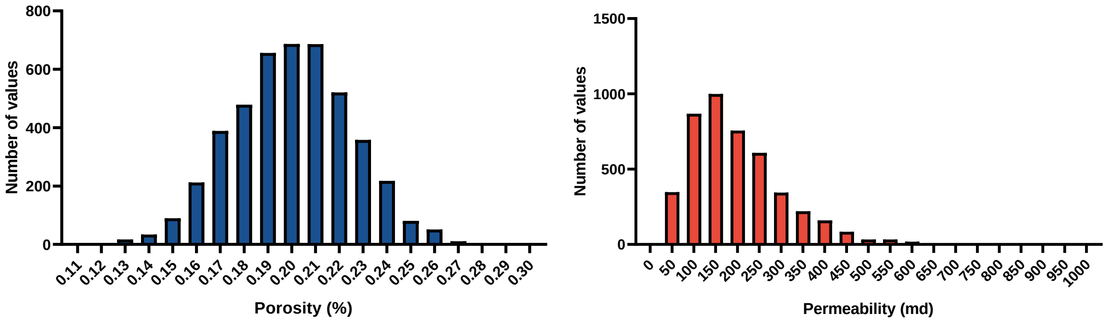


Figure 5, the porosity and permeability distribution for case study 1

Table 5, Reservoir configurations for case study 1

Properties	Reservoir Description
Number of Cells in the X and Y direction	30
Number of Cells in the Z direction	5
X, Y, Z-Direction Cell Size (ft)	300, 100, 300
Mean Porosity (%)	20
Mean Permeability (md)	200
Reservoir Depth (ft)	5000
Rock Compressibility @ Ref Pressure (1/psi)	0.0000045
Gas Density (lb/ft^3)	0.043
PVT Table	PVT 1
Initial Water Saturation (%)	2
Initial Pressure (psi)	2325
Well Location	X=15, Y=5
Perforation Intervals	Z= 1:2
Well Rate (MSCF/d)	120000
Bottom Hole Pressure (psi)	1200
Time Step (days)	15
Production Duration (days)	600

Table 6 compares the DNS and ECLIPSE models using different metrics. As can be seen, the results for both cells have similar accuracy, even though they were in the different parts of the reservoir. DNS predictions have MAE of 4 psi, MAPE of almost zero, and R2 of almost 1. These excellent values for each metric illustrate how DNS was able to calculate the correct pressure change across the reservoir.

Table 6, The accuracy of DNS in predicting the pressure across the reservoir for case study 1

Cell Number	MAE (psi)	MAPE (%)	MRE (psi)	MSE	RMSE	R2
Cell 1	4.505	0.203	0.002	23.829	4.882	0.999
Cell 2	4.505	0.203	0.002	23.829	4.882	0.999

To study these results with a greater level of detail, we evaluated two cells and thoroughly examined them. One of the selected cells contained the wellbore, and the other was a randomly selected cell far away from the wellbore.

Figure 6a demonstrates the pressure changes of the cell with 20 x 20 x 4 coordinates over 600 days; this was calculated with both DNS and ECLIPSE. As can be seen, DNS was able to produce almost exactly the same results as the ECLIPSE simulation. In addition, DNS was able to predict the change of slope that occurred within the last quarter of the production time successfully. Figure 6c displays the relative error (RE) and the relative percentage error (RPE) of different timesteps for this cell. From this figure, it can be observed that the maximum value for MAE is around 9 psi.

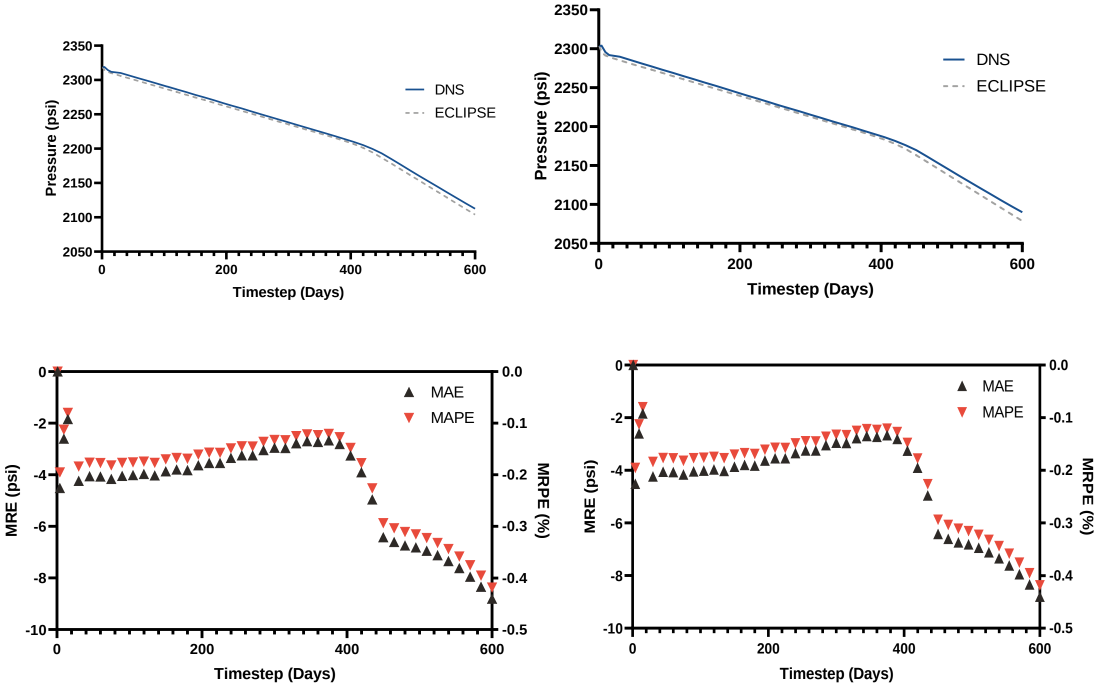


Figure 6, Pressure trend, RE and RPE for the cell 15 x 5 x 1 (left) and the cell 20 x 20 x 4 (right)

Figure 6b shows the pressure change of the cell containing the wellbore. Like with the previous cell, both trends are highly similar. Interestingly, the trend of RE and RPE for this cell are similar to the trend of the other cell (Figure 6c and Figure 6d).

The pressure distribution in the reservoir over 12 steps is then compared. Figure 7 compares the pressure distribution in the XY plane calculated by DNS and ECLIPSE. According to these figures, DNS predicted that the transition flow lasts for five timesteps, whereas ECLIPSE predicted that this flow lasts for four timesteps. In both simulators, the rates of pressure drop in pseudo-steady state flow are the same, confirming the precision of DNS, as even such a trivial feature was correctly predicted.

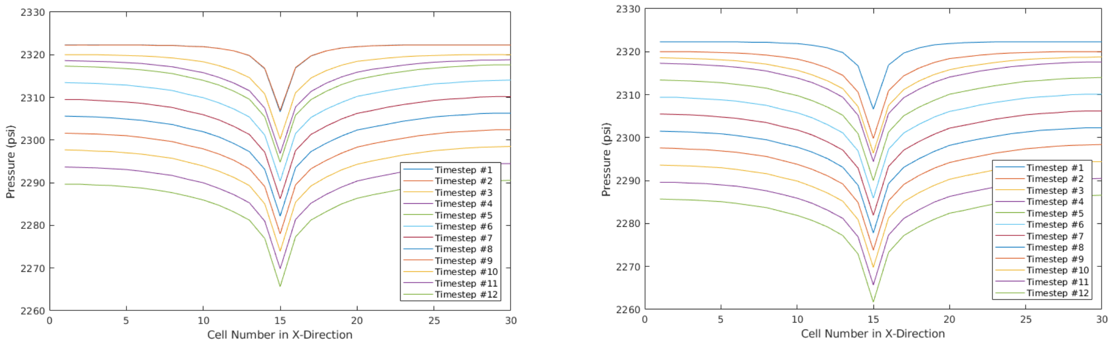


Figure 7, Pressure distribution throughout the reservoir for different time steps using DNS (left) and a commercial simulator (right)

Next, we compare the pressure distribution across the reservoir. Figure 8 illustrates a snapshot of pressure distribution for two time steps. Figures on the left were plotted using the results from DNS and the rest were plotted using results obtained from ECLIPSE.

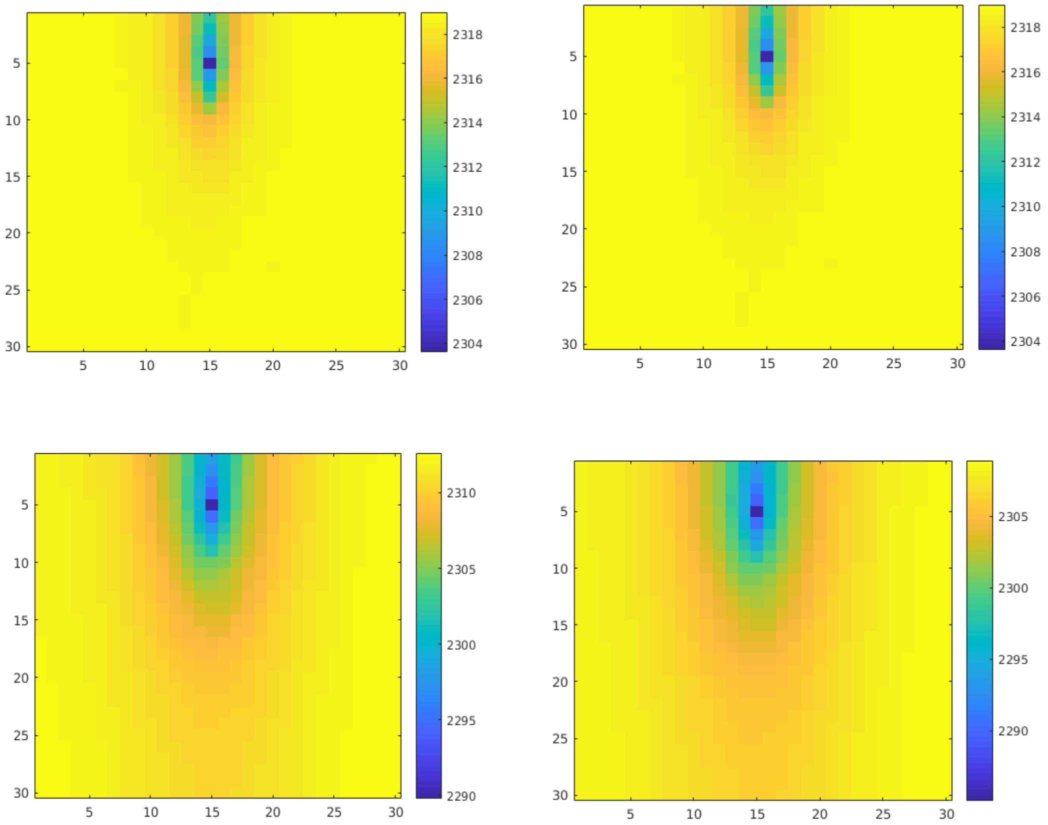


Figure 8, Pressure map throughout the reservoir for two time steps using DNS (left) and a commercial simulator (right)

When comparing the results obtained from DNS with the ECLIPSE model, a high level of similarity was noted. This suggests that DNS can predict the outcomes of this case study with a high degree of accuracy.

Case study 2

In this case, a couple of the parameters, such as gas density, initial pressure, well rate and BHP, were slightly altered. First, it consisted of a smaller, but higher amount of grid cells: 12,500 cells, and a new set of PVT data. This change allows us to see how DNS reacts to a completely different reservoir fluid. The production duration for this case is still two years, but a higher well rate is achieved. Table 7 summarises the reservoir description for this case study. Figure 9 illustrates the porosity and permeability distribution, respectively.

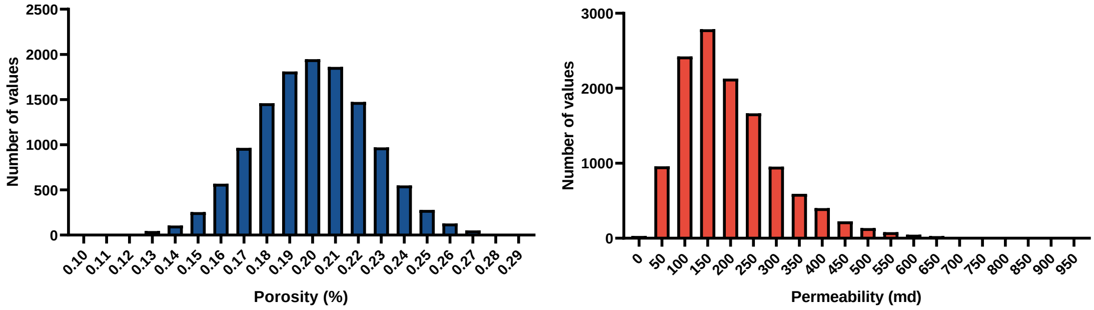


Figure 9, The porosity and permeability distribution for case study 2

Table 7, Reservoir configurations for case study 2

Properties	**Reservoir Description**
Number of Cells in the X and Y direction	50
Number of Cells in the Z direction	5
X, Y, Z-Direction Cell Size (ft)	100, 100, 200
Mean Porosity (%)	20
Mean Permeability (md)	202
Reservoir Depth (ft)	5500
Rock Compressibility @ Ref Pressure (1/psi)	0.0000042
Gas Density (lb/ft^3)	0.045
PVT Table	PVT 2
Initial Water Saturation (%)	22
Initial Pressure (psi)	2558
Well Location	X=10 , Y=20
Perforation Intervals	Z= 1:5
Well Rate (MSCF/d)	200000
Bottom Hole Pressure (psi)	500
Time Step (days)	15
Production Duration (days)	600

Once again, we evaluated two cells. One cell was connected to the wellbore at coordinates 10 x 20 x 1, the other cell was located far away from the wellbore at 30 x 30 x 3. As indicated by Table 58, DNS was able to predict the pressure drop accurately. The low value of MAE and MAPE present in this case indicates that the flow behaviour was predicted with accuracy. Figure 10a and Figure 10b illustrate the pressure drops calculated with both DNS and ECLIPSE which are almost identical.

Table 58, Accuracy of DNS in predicting the pressure across the reservoir for case study 2

Cell number	MAE (psi)	MAPE (%)	MRE (psi)	MSE	RMSE	R2
Cell 1	3.154	0.138	0.000	38.169	6.178	1.000
Cell 2	3.154	0.138	0.000	38.169	6.178	1.000

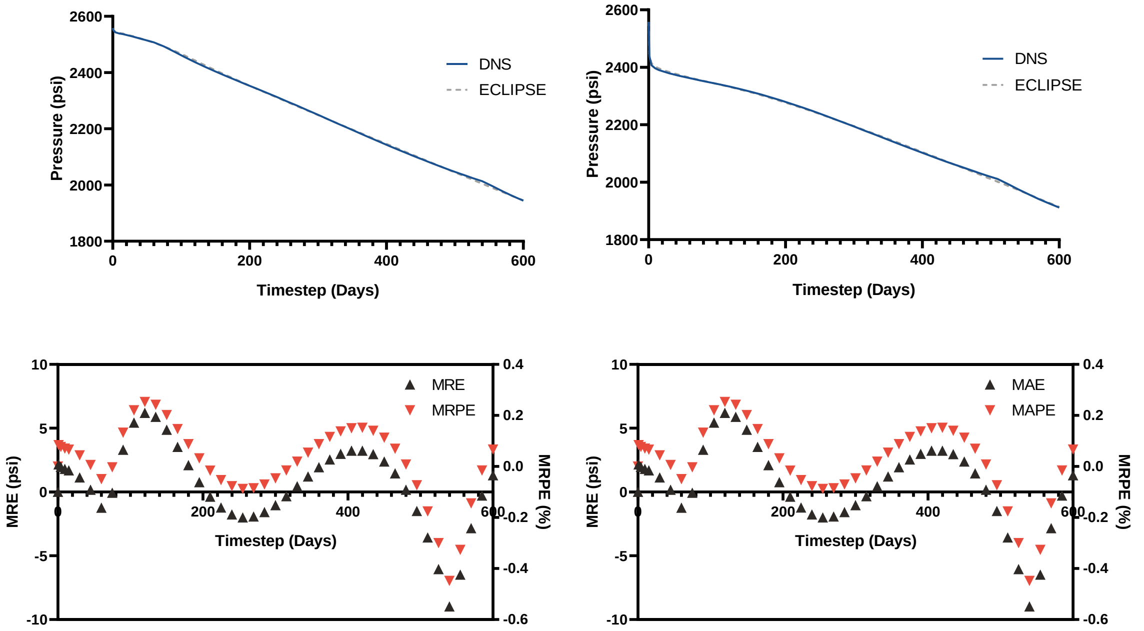


Figure 10, Pressure trend, RE and RPE for cell 10 x 20 x 1 (left) and cell 30 x 30 x 3 (right)

Figure 10c and Figure 10d display the RE and PRE throughout the production time. It can be observed that the RE values vary between -10 to 7 psi, which is negligible.

The pressure drop that occurred within the cells with wellbores were then examined. The value of these predicted errors was the same as that of other previously-analysed cells, Table 58. Like the other cells, this pressure change was accurately predicted by DNS when compared with ECLIPSE. Furthermore, the results obtained for RE and RPE distribution in these two cells are of high similarity.

Figure 11 compares the pressure distribution on the wellbore plane within the reservoir. This was calculated with both DNS and ECLIPSE. Like in the previous case, DNS was able to predict the pressure distribution correctly.

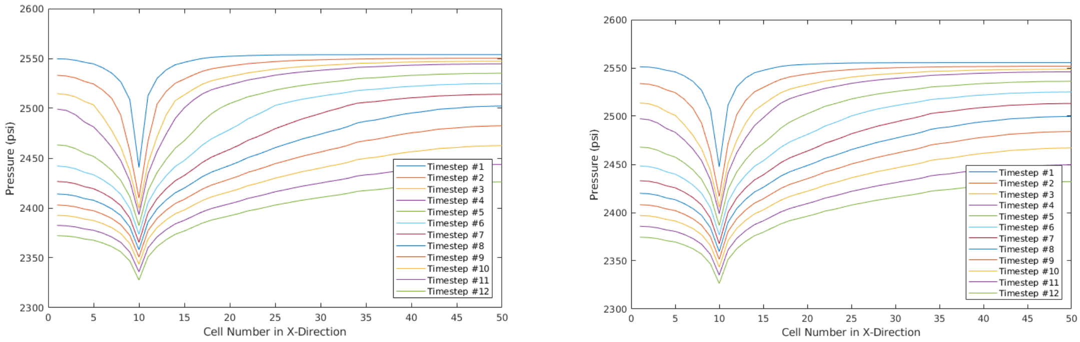


Figure 11, Pressure distribution throughout the reservoir for different time steps using DNS (left) and a commercial simulator (right)

Figure 12 indicates the pressure map on the well plane for two time steps. Again, DNS was able to predict the pressure distribution over the production time and throughout the reservoir correctly.

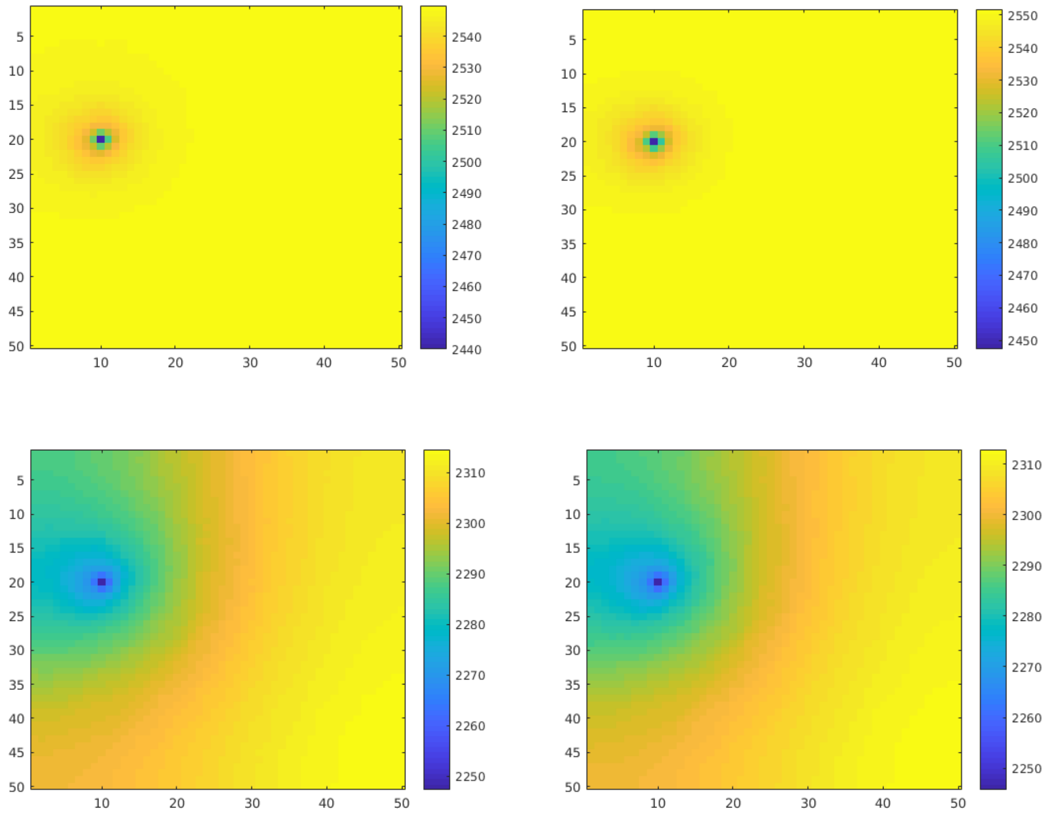


Figure 12, Pressure map throughout the reservoir for two time steps using DNS (left) and a commercial simulator (right)

Case study 3

Table 9 lists the reservoir description for a reservoir with 4500 cells. This case has the lowest well rate, porosity and permeability among all the cases. This well is perforated only at the upper cells, with a BHP of 1000 psi, porosity equal to 16 and a horizontal permeability of 78md. Figure 13 shows the porosity and permeability distribution for this case study.

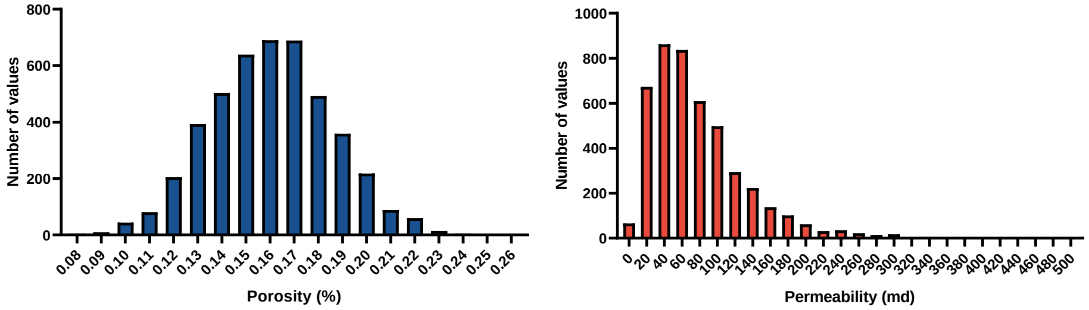


Figure 13, The porosity and permeability distribution for case study 3

Table 9, Reservoir configurations for case study 3

Properties	**Reservoir Description**
Number of Cells in the X and Y direction	30
Number of Cells in the Z direction	5
X, Y, Z-Direction Cell Size (ft)	300, 300, 100
Mean Porosity (%)	16
Mean Permeability (md)	78
Reservoir Depth (ft)	5000
Rock Compressibility @ Ref Pressure (1/psi)	0.0000042
Gas Density (lb/ft^3)	0.043
PVT Table	PVT 2
Initial Water Saturation (%)	20
Initial Pressure (psi)	2325
Well Location	X=5 , Y=10
Perforation Intervals	Z= 1:1
Well Rate (MSCF/d)	20000
Bottom Hole Pressure (psi)	1000
Time Step (days)	10
Production Duration (days)	400

Since previous cases show consistent errors across all cells, the following cases report the results of the cell with the wellbore and compare its predicted values with that of ECLIPSE’s output. Similar to the previous cases, the metrics are shown in

Table 10 and highlight the accuracy of DNS in predicting the pressure change in this reservoir.

Table 10, The accuracy of DNS in predicting the pressure across the reservoir for case study 3

MAE (psi)	MAPE (%)	MRE (psi)	MSE	RMSE	R2
5.635	0.256	-0.003	676.736	26.014	0.993

Figure 14a compares the pressure changes of these cells throughout the production phase. Figure 14b illustrates the change of RE and PRE over the predetermined period. Both of these figures indicate that DNS was able to represent the phenomenon of the gas flow in the porous media.

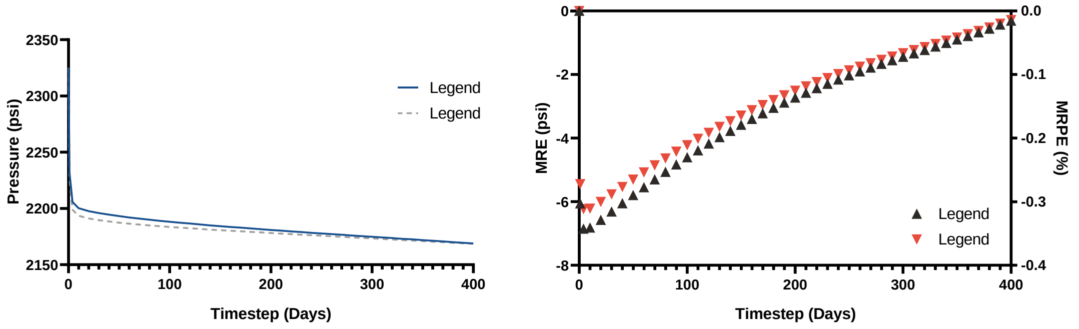


Figure 14, Pressure trend, RE and RPE for the cell connected to the wellbore

As illustrated in Figure 15, DNS understands the change of the flow patterns correctly. In these figures, it is obvious that DNS can predict the transient flow properly and determine the time taken for the flow to reach the outer boundary. Furthermore, the change of flow regime from transition to pseudo-steady state is predicated in a similar way to the outcome from ECLIPSE.

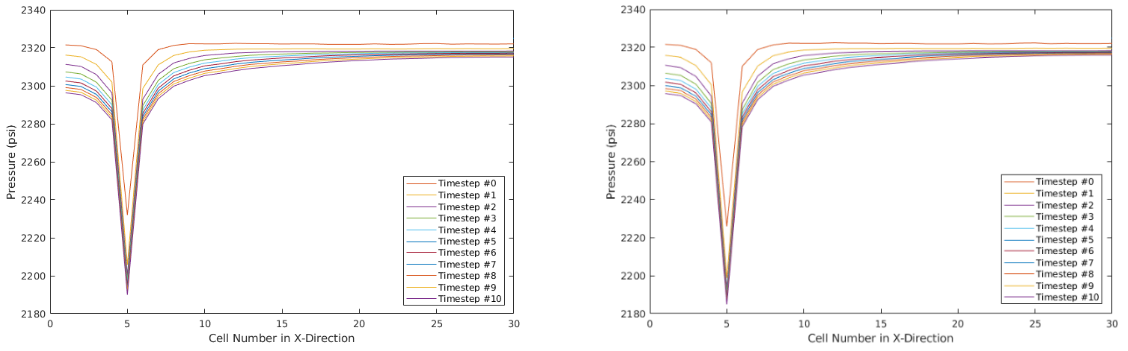


Figure 15, Pressure distribution throughout the reservoir for different time steps using DNS (left) and a commercial simulator (right)

The pressure maps for two timesteps are illustrated in Figure 16. In this Figure, the top and bottom snapshots are associated with time step #3 and time step #35, respectively. These figures confirm the accuracy displayed by DNS in the prediction of the pressure distribution throughout the reservoir. As shown in these figures, the front lines in these figures are predicted like the results from ECLIPSE in the right figures.

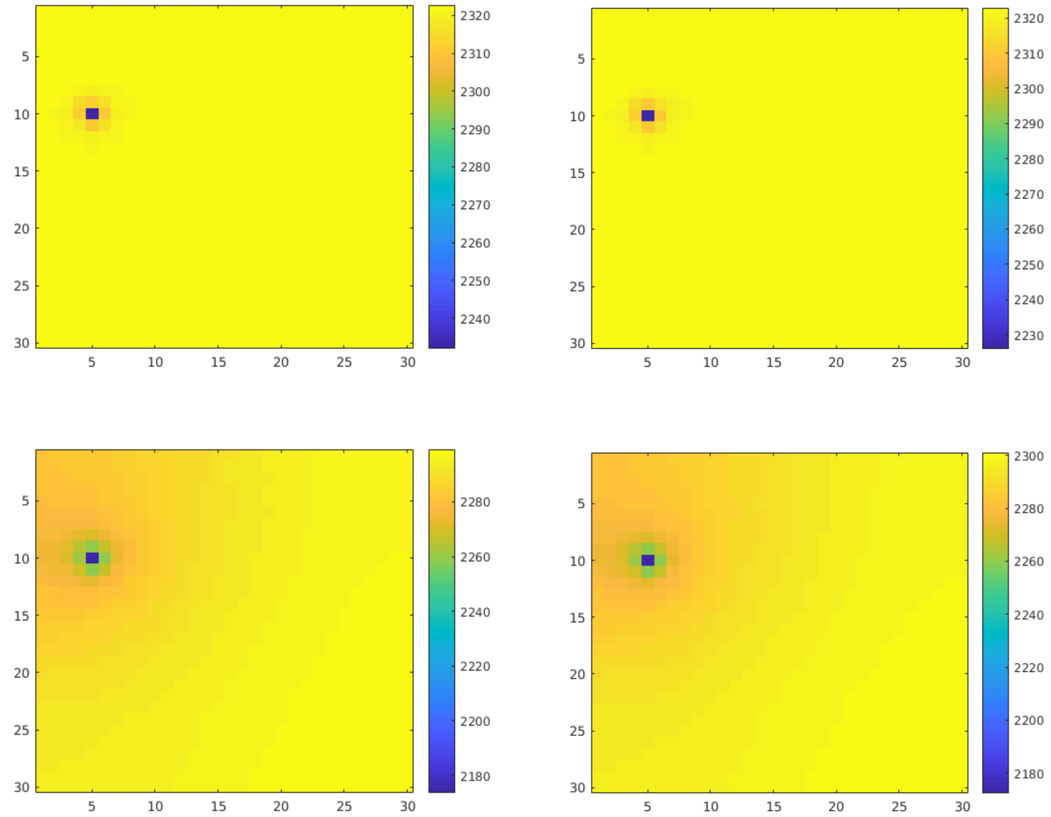


Figure 16, Pressure map throughout the reservoir for two time steps using DNS (left) and a commercial simulator (right)

Case study 4

Case study 4 was observed to be the case with the highest well rate presented in this section. This well was partially completed, with field PVT data being used in this case to investigate if DNS has captured the essence and physics of the fluid flow. There is a significantly higher well rate assigned to this case in comparison to previous cases. Table 11 shows the complete description of this case study. Similar to the previous case, the porosity and permeability distribution are displayed in Figure 17. The presence of low error values and high correlation coefficient values indicates that DNS was able to predict the output values correctly (Table 12). Figure 18a demonstrations that DNS is able to predict the same trend as the ECLIPSE model. Figure 18b shows the accuracy of the DNS prediction over the production time.

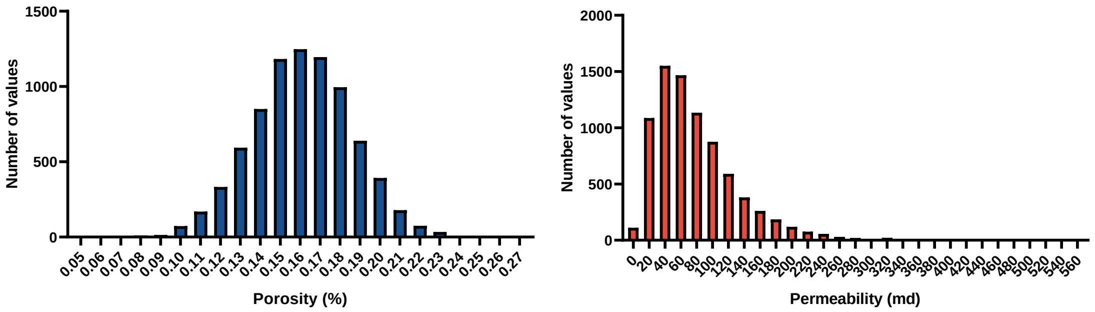


Figure 17, The porosity and permeability distribution for case study 4

Table 11, Reservoir configurations for case study 4

Properties	**Reservoir Description**
Number of Cells in the X and Y direction	40
Number of Cells in the Z direction	5
X, Y, Z-Direction Cell Size (ft)	60, 200, 100
Mean Porosity (%)	16
Mean Permeability (md)	80
Reservoir Depth (ft)	6000
Rock Compressibility @ Ref Pressure (1/psi)	0.0000045
Gas Density (lb/ft^3)	0.042
PVT Table	PVT 3
Initial Water Saturation (%)	20
Initial Pressure (psi)	2790
Well Location	X=10, Y=5
Perforation Intervals	Z= 1:3
Well Rate (MSCF/d)	300000
Bottom Hole Pressure (psi)	1000
Time Step (days)	10
Production Duration (days)	600

Table 12, The accuracy of DNS in predicting the pressure across the reservoir for case study 4

MAE (psi)	MAPE (%)	MRE (psi)	MSE	RMSE	R2
15.391	0.710	0.006	590.623	24.303	0.994

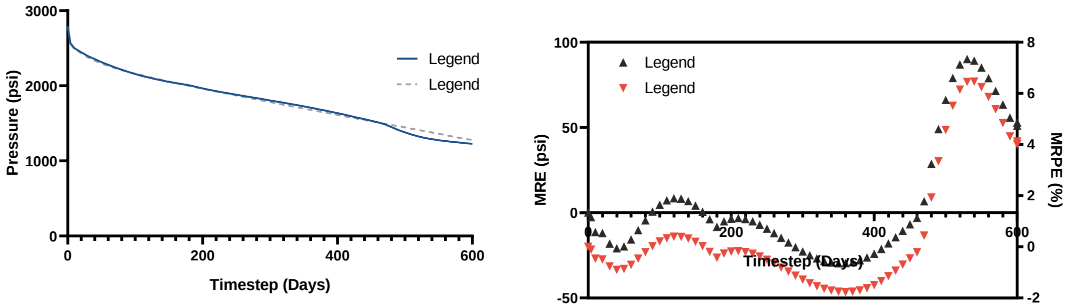


Figure 18, Pressure trend, RE and RPE for the cell connected to the wellbore

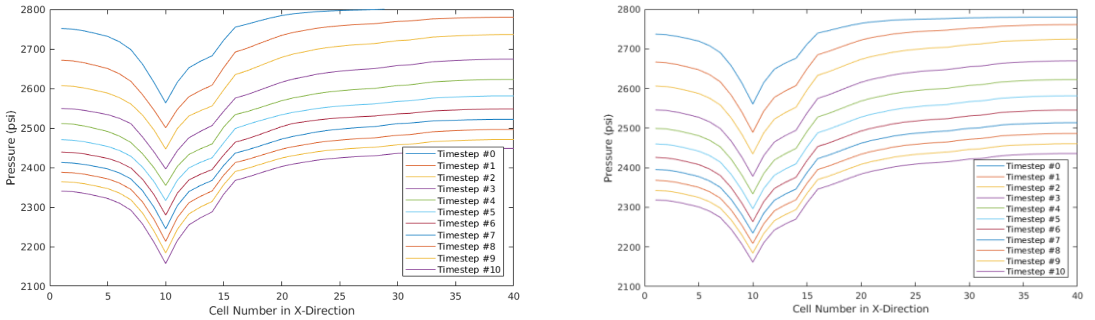


Figure 19, Pressure distribution throughout the reservoir for different time steps using DNS (left) and a commercial simulator (right)

As can be observed in Figure 19, the results obtained by DNS and ECLIPSE respectively have extremely high similarity. However, the time taken to produce such plots varied drastically, with DNS producing the above plot in the span of less than a second.

Figure 20 demonstrates the pressure distribution in a 2D plane over time. It can be observed that DNS was able to predict this distribution flawlessly, with the two pressure maps appearing almost identical.

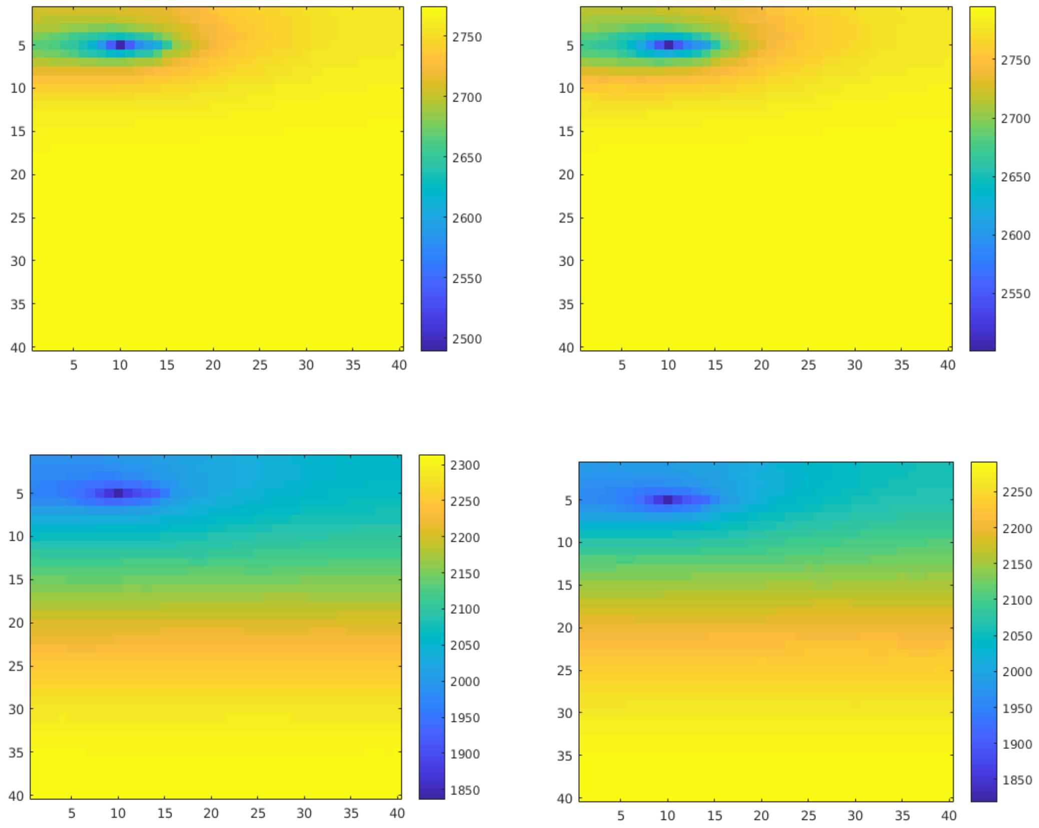


Figure 20, Pressure map throughout the reservoir for two time steps using DNS (left) and a commercial simulator (right)

To further examine the DNS, Table 513 was produced to represent more scenarios, to better validate the accuracy of DNS. In this table, cases with various configurations compared with the aforementioned case studies are presented. Table 14 provides a comparison of DNS and ECLIPSE for these reservoirs. Yet again, the results obtained through DNS were remarkably accurate compared with those from ECLIPSE.

Table 513, Reservoir configurations for case studies 5 to 9

Properties	Case 5	Case 6	Case 7	Case 8	Case 9
Number of Cells in the X and Y direction	40	50	50	30	40
Number of Cells in the Z direction	5	5	5	5	5
X, Y, Z-Direction Cell Size (ft)	200-100-100	200-200-100	100-200-100	100-200-100	100-100-100
Mean Porosity (%)	21	24	22	17	24
Mean Permeability (md)	248.5		301	102	440
Reservoir Depth (ft)	9500	9500	8000	7000	7500
Rock Compressibility @ Ref Pressure (1/psi)	3.2	3.5	3.5	3.5	3.5
Gas Density (lb/ft^3)	0.044	0.046	0.043	0.045	0.047
PVT Table	2	1	3	2	3
Initial Water Saturation (%)	0.2	0.24	0.2	0.22	0.2
Initial Pressure (psi)	4418	4418	3720	3255	3488
Well Location	5–10	10–10	5-5	10-15	5-5
Perforation Intervals	1-3	1-3	1-5	1-2	1-5
Well Rate (MSCF/d)	20000	200000	200000	150000	150000
Bottom Hole Pressure (psi)	1250	900	1200	1200	950
Time Step (days)	10	10–10	10	10	20
Production Duration (days)	400	300	400	400	800

Table 14, The accuracy of DNS in predicting the pressure across the reservoir for case studies 5 to 9

Case	MAE (psi)	MAPE (%)	MRE (psi)	MSE	RMSE	R2
5	4.953	0.113	-0.001	49.618	7.0440	0.884
6	34.539	0.795	-0.001	1658.12	40.72	0.9645
7	53.587	1.41253	0.01395	13290.6	115.285	0.98298
8	46.1246	1.39781	0.00611	18396.8	135.635	0.98298
9	42.0364	1.18225	0.01123	20316	142.534	0.98298

- **Speed Results:**

To analyse the speed of DNS, we designed several cases of varying sizes. Table 15 lists these case studies and Table 16 reports the observed accuracy of DNS in comparison with ECLIPSE. In all cases, it can be observed that DNS was able to predict the pressure changes over the production time accurately. A major advantage of DNS is that it does not need to be run across every cell, unlike commercial software. This requirement for commercial software originates from the fact that it takes into account the initial and boundary conditions directly through numerical methods. As a result, DNS only needs to be used for a handful of selected cells, gaining significant acceleration for the simulation.

As has been shown throughout the previous case studies, DNS tends to show the same errors in all the cells within a specific reservoir. This allows us to run DNS for the desired grid cells (the ones connected to the wellbore) alone, although the times taken to run DNS for all grid cells are presented, regardless.

Another advantage of DNS is that it can be used to estimate only the desired production period. This means that, after defining the initial conditions, we can run a simulation, for example, starting two years after the recorded initial time. This flexibility assists us in accelerating the simulation. Another advantage occurs when we deal with physics-based reservoir models that have convergence problems when solved with numerical approaches. This convergence problem takes the simulation longer. Unlike numerical simulations, DNS avoids this problem as it is a data-driven model, containing an explicit mathematic formula that is able to estimate the desired outputs.

Table 17 shows the runtime of the aforementioned case studies. The first row of the table displays the time taken by ECLIPSE to simulate these models. Obviously, the speed of the simulation directly depends on the number of cells and timesteps present. Since ECLIPSE solves a series of partial differential equations, all cells within the boundary must be considered. Therefore, for example, in the second case, it took ECLIPSE more than 13 minutes to solve the pressure change. However, due to DNS being developed independently of the boundary and initial conditions, only the desired cells needed to be simulated, which in this case were the cells connected to the wellbore (three grid cells). It took DNS fractions of a second (9.00E-05 seconds) to simulate the pressure change over the production period for each cell, meaning it takes 2.70E-04 seconds to simulate all three cells. This is a %2.90E+08 reduction in the runtime when compared with ECLIPSE. Furthermore, if we simulated the entire reservoir grid, DNS could still reduce the runtime by 443%. As can be seen in this table, the amount of runtime reduction is significant, particularly when we deal with a large-scale study. Overall, in the six large-scale case studies that we investigated in this paper, on average, DNS was able to reduce the runtime by 9.25E+07% if we simulate only the cells connected to the wellbore and 117% if we simulate the entire grid cell.

Table 15, Reservoir configurations for large-scale case studies

Properties	Case 10	Case 11	Case 12	Case 13	Case 14	Case 15
Grid size	2,500,000	1,600,000	2,500,000	900,000	1,000,000	16,000
X, Y, Z-Direction Cell Size (ft)	50-60-100	80-50-50	100-200-50	50,50,50	100,100,100	100,100,100
Mean Porosity (%)	18	19	21	16	22	14
Mean Permeability			247	78	300	
Reservoir Depth (ft)	5000	6000	5000	7000	4000	8000
Rock Compressibility @ 3000 psi (1/psi)	3.5e-6	3.5e-6	3.5e-6	3.5e-6	3.5e-6	3.5e-6
Gas Density (lb/ft^3)	0.047	0.041	0.042	0.04	0.045	0.04
PVT Table	1	2	1	3	2	3
Initial Water Saturation (%)	20	20	25	21	22	21
Initial Pressure (psi)	2325	2790	2325	3255	1860	3720
Well Location	5-5	5-5	5-15	250-250	50,50	25,25
Perforation Intervals	1-5	1-3	1-5	1-5	3-5	1-5
Well Rate (MSCF/d)	400000	120000	150000	600,000	150,000	200,000
Bottom Hole Pressure (psi)	700	1200	700	1200	1000	1200
Time Step (days)	10	15	30	15	20	15
Production Duration (days)	500	900	1200	1800	2000	1500

Table 16, The accuracy of DNS in predicting the pressure across the reservoir for case studies with a large number of grid cells

Case	MAE (psi)	MAPE (%)	MRE (psi)	MSE	RMSE	R2
10	8.244	0.395	-0.003	519.015	22.782	1.000
11	9.725	0.373	-0.002	269.216	16.408	0.997
12	4.105	0.188	0.002	509.361	22.569	1.000
13	9.893	0.514	-0.003	726.621	26.956	1.000
14	10.698	0.522	0.004	764.042	27.641	1.000
15	12.838	0.574	0.005	687.638	26.223	1.000

Table 17, Comparing the Runtime of DNS and ECLIPSE

Properties	Case 10	Case 11	Case 12	Case 13	Case 14	Case 15
ECLIPSE runtime (Sec)	390	782	188	328	185	2.4
DNS runtime (Sec /Cell/Time Step)	1.50E-06	1.50E-06	1.50E-06	1.50E-06	1.50E-06	1.50E-06
DNS runtime (Sec /Cell)	7.50E-05	9.00E-05	6.00E-05	1.80E-04	1.50E-04	1.50E-04
DNS Runtime (Sec)	3.75E-04	2.70E-04	3.00E-04	9.00E-04	3.00E-04	7.50E-04
Runtime Percentage reduction (%) – target cells	1.04E+08	2.90E+08	6.27E+07	3.64E+07	6.17E+07	3.20E+05
DNS runtime (Sec) – full grid	187.50	144.00	150.00	162.00	150.00	2.40
Runtime Percentage reduction (%) – entire grid	108.00	443.06	25.33	102.47	23.33	0.00

Figure 21 represents the runtime of cases 10 and 11, simulated using DNS and ECLIPSE, when we run each model multiple times. As can be seen in this Figure, across 1000 simulations, we can save 56 and 177 hours for cases 10 and 11, respectively, if we use DNS instead of ECLIPSE. This is equal to a 108% and 443% reduction on the runtime.

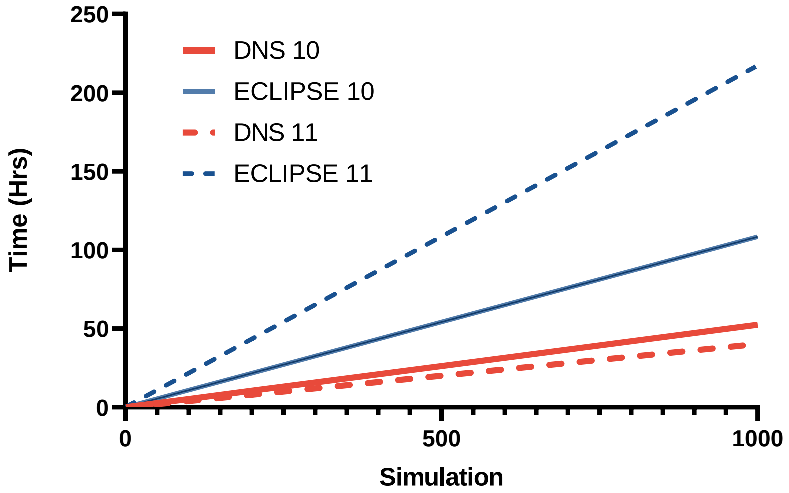


Figure 21, Comparing the runtime of case studies 10 and 11 for multiple simulations

- **Reliability**

One of the main concerns in any machine learning outcome is its reliability. Although machine learning methods can provide some remarkable results with any data, they may behave completely bizarrely over some other data. Therefore, throughout this research, we ran DNS for about 600 case studies, overall containing more than 500,000,000 data points and they were compared against the results obtained from a commercial simulator. Figure 22 displays a histogram (left) and a cumulative histogram (right), showing the accuracy given by DNS for every unseen case study. For approximately 45% of the simulated cases, DNS displays incredible accuracy, with a MAPE less than 5%. Additionally, over 70% of all cases were observed to have a MAPE less than 10%, and 90% have a MAPE less than 15%. Although in most of the cases that were studied, DNS behaved with incredible accuracy. There were a couple of cases where the results were not satisfactory. This implies that even with this approach, which we believe has a huge potential considering its formidable speed, this current version of DNS can be further improved, enabling higher reliability. We believe DNS can be deemed reliable if it is able to predict all cases with an overall error of less than 5%. Given that even commercial simulators suffer from huge mismatches when compared with real production data, DNS can be used reliably, once this objective has been achieved.

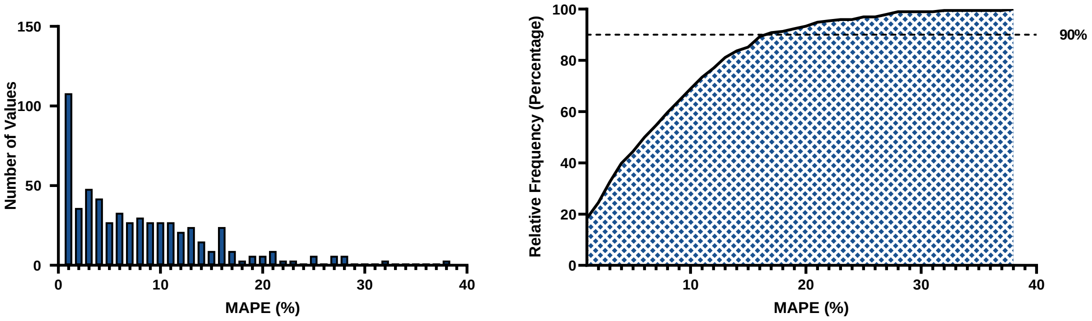


Figure 22, The histogram of the accuracy of DNS for all the case studies in this study

- **Limitations**

Despite the accuracy that DNS has provided throughout these benchmark cases, there are still limitations to be considered.

The first is that we only consider gas as the fluid flow in this study. Considering there are different physics governing gas and liquid flows, we had to differentiate between these two studies. We decided to focus on gas flow on the current research and investigate liquid flow in a future study.

The second limitation revolves around only a single vertical production well being considered within the reservoir. The main reason we opted to start with a vertical well was that we are at the primary phase of this ongoing study. In the next phase, we will investigate the effect of multiple wellbores on the accuracy of DNS. However, it must be noted that the results from this study are still valid, as wellbores are not drilled with the same drainage area.

The third limitation of this approach revolves around the lack of different geological structures attempted. This was mainly due to the lack of computational power. In order to make a comprehensive data set that covers a proper range of geological structures, we would need a ten times bigger dataset than the one we used in this study. Training such an enormous dataset over a deep network requires massive computational power. This was not an option during the primary phase of this ongoing study; however, we have planned to improve the computational power for the next phase of this study.

Regardless of these limitations, we believe that the produced results have great potential. Furthermore, by increasing the range of training data while improving the computational power could minimise these limitations. Therefore, we believe a pre-trained data-driven simulator that is capable of predicting the behaviour of the fluid within any reservoir can eventually be built using the approach presented in this study.

Conclusion

This paper describes the logistics of designing and implementing a data-driven model, capable of predicting the pressure across the draining area of a vertical well within a natural gas reservoir. The novelty of this study lies within the implementation of a deep learning network to develop a trained model, which provides a comprehensive tool as a fast data-driven replacement for a commercial simulator. The proposed model can predict reservoir behaviour for a wide range of reservoir configurations.

The results obtained confirm that this approach has the potential to be used in building a pre-trained data-driven simulator, capable of providing accurate, fast and reliable results for any given reservoir configuration. Unlike conventional approaches to proxy modelling, DNS is not built for only one specific reservoir. On the contrary, DNS is built to be used in predicting reservoir performance for a wide range of reservoir configurations, without requiring further training or building processes.

We evaluated this model in terms of precision, speed and reliability. These results were then compared with a commercial simulator, Schlumberger ECLIPSE. 600 different synthetic reservoirs were generated, each with various descriptions and production settings. We used multiple metrics to evaluate these case studies; MAE, MRE, MAPE, MSE, RMSE and R2.

We presented the precision of DNS, by comparing its results with ECLIPSE in nine case studies. In each case study, we compared the pressure drop over time, pressure distribution across the reservoir and the pressure map produced by DNS and ECLIPSE. In all cases, DNS was able to learn the fluid behaviour accurately and thus predict the pressure with remarkable precision.

We then investigated six case studies with larger grid sizes, each containing a different number of cells. Due to the indirect dependency of DNS on initial and boundary conditions, DNS can directly simulate specific cells within a particular period. This allows DNS to be incredibly fast when compared with a physics-based simulator. Therefore, it takes DNS only 1.5e-6 seconds to predict the cell pressure for one timestep. Through this approach, it takes DNS on average 1.18E-04 seconds to predict the pressure trend of the wellbore for five case studies. This means that, on average, DNS is 2e10 times faster than a commercial simulator. Furthermore, it must be noted, if we want to simulate the entire grid, DNS would still be comparatively faster, specifically when the numerical model has a convergence problem. We showed this approach could reduce the runtime by 108% and 443% (56 and 177 hours) when simulating two of the large-scale case studies 1000 times.

Finally, we investigated the reliability of DNS when predicting 600 benchmark cases, containing 500,000,000 data points. DNS was able to predict 45%, 70% and 90% of the cases with less than 5%, 10% and 15% MAPE, respectively. Given the speed of DNS, these results are significantly better than a commercial simulator.

This ongoing study aims to develop a data-driven model that is able to act as a proxy model for numerical reservoir simulators. This simulator is aimed to be used to investigate a variety of scenarios quickly during history matching, uncertainty quantification and field optimisation, allowing for quicker and better decision-making processes.

Ethics declarations:

Conflict of interest

The authors confirm that there are no known conflicts of interest associated with this publication.

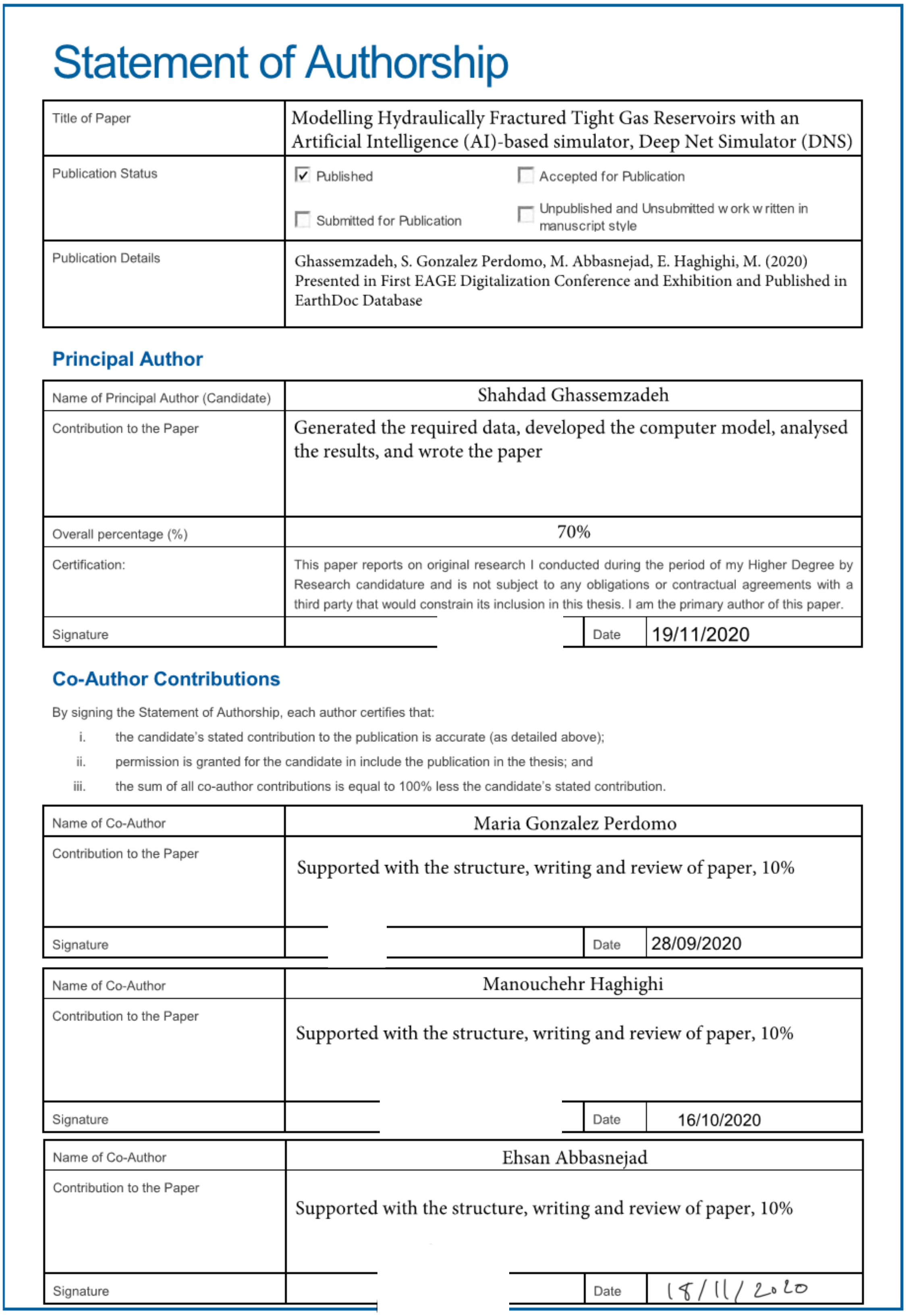
References

1. Ertekin T, Abou-Kassem JH, King GR (2001) Basic applied reservoir simulation. vol Sirsi) i9781555630898.
2. Jansen J-D, Brouwer R, Douma SG Closed loop reservoir management. In: SPE reservoir simulation symposium, 2009. Society of Petroleum Engineers,
3. Beckner BL, Haugen KB, Maliassov S, Dyadechko V, Wiegand KD (2015) General Parallel Reservoir Simulation. Paper presented at the Abu Dhabi International Petroleum Exhibition and Conference, Abu Dhabi, UAE, 2015/11/9/
4. Eldred ME, Orangi A, Al-Emadi AA, Ahmad A, O'Reilly TJ, Barghouti N (2014) Reservoir Simulations in a High Performance Cloud Computing Environment. Paper presented at the SPE Intelligent Energy Conference & Exhibition, Utrecht, The Netherlands, 2014/4/1/
5. Coats KJJoPT (1969) Use and misuse of reservoir simulation models. 21 (11):1,391-391,398
6. Fanchi JR (2001) Principles of applied reservoir simulation. 2nd edn. Gulf Pub., Boston
7. Mohaghegh SD (2017) Data-Driven Reservoir Modeling. Society of Petroleum Engineers,
8. Abooali D, Khamehchi E (2019) New predictive method for estimation of natural gas hydrate formation temperature using genetic programming. Neural Computing and Applications 31 (7):2485-2494. doi:10.1007/s00521-017-3208-0
9. Jensen JL (2017) Comment on “Characterising interwell connectivity in waterflooded reservoirs using data-driven and reduced-physics models: a comparative study” by E. Artun DOI 10.1007/s00521-015-2152-0. Neural Computing and Applications 28 (7):1745-1746. doi:10.1007/s00521-016-2549-4
10. Zheng J, Leung JY, Sawatzky RP, Alvarez JM (2019) An AI-based workflow for estimating shale barrier configurations from SAGD production histories. Neural Computing and Applications 31 (9):5273-5297. doi:10.1007/s00521-018-3365-9
11. Artun E, Ertekin T, Watson R, Al-Wadhahi M (2011) Development of universal proxy models for screening and optimisation of cyclic pressure pulsing in naturally fractured reservoirs. Journal of Natural Gas Science and Engineering 3 (6):667-686. doi:https://doi.org/10.1016/j.jngse.2011.07.016
12. Ghassemzadeh S, Charkhi AH (2016) Optimisation of integrated production system using advanced proxy based models: A new approach. Journal of Natural Gas Science and Engineering 35:89-96. doi:https://doi.org/10.1016/j.jngse.2016.08.045

13. Kalantari-Dahaghi A, Mohaghegh S, Esmaili S (2015) Data-driven proxy at hydraulic fracture cluster level: A technique for efficient CO2- enhanced gas recovery and storage assessment in shale reservoir. Journal of Natural Gas Science and Engineering 27:515-530. doi:https://doi.org/10.1016/j.jngse.2015.06.039
14. Chen B, He J, Wen X-H, Chen W, Reynolds AC (2017) Uncertainty quantification and value of information assessment using proxies and Markov chain Monte Carlo method for a pilot project. Journal of Petroleum Science and Engineering 157:328-339. doi:https://doi.org/10.1016/j.petrol.2017.07.039
15. He J, Xie J, Wen X-H, Chen W (2016) An alternative proxy for history matching using proxy-for-data approach and reduced order modeling. Journal of Petroleum Science and Engineering 146:392-399. doi:https://doi.org/10.1016/j.petrol.2016.05.026
16. Kim M, Shin H (2018) Development and application of proxy models for predicting the shale barrier size using reservoir parameters and SAGD production data. Journal of Petroleum Science and Engineering 170:331-344. doi:https://doi.org/10.1016/j.petrol.2018.06.044
17. Nwachukwu A, Jeong H, Pyrcz M, Lake LW (2018) Fast evaluation of well placements in heterogeneous reservoir models using machine learning. Journal of Petroleum Science and Engineering 163:463-475. doi:https://doi.org/10.1016/j.petrol.2018.01.019
18. Goodwin N (2015) Bridging the Gap Between Deterministic and Probabilistic Uncertainty Quantification Using Advanced Proxy Based Methods. Paper presented at the SPE Reservoir Simulation Symposium, Houston, Texas, USA, 2015/2/23/
19. Alenezi F, Mohaghegh S (2017) Developing a Smart Proxy for the SACROC Water-Flooding Numerical Reservoir Simulation Model. Paper presented at the SPE Western Regional Meeting, Bakersfield, California, 2017/4/23/
20. Alenezi F, Mohaghegh S A data-driven smart proxy model for a comprehensive reservoir simulation. In: 2016 4th Saudi International Conference on Information Technology (Big Data Analysis)(KACSTIT), 2016. IEEE, pp 1-6
21. Mohaghegh SD, Gaskari R, Maysami M (2017) Shale Analytics: Making Production and Operational Decisions Based on Facts: A Case Study in Marcellus Shale. Paper presented at the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, 2017/1/24/
22. Ghassemzadeh S, Perdomo MG, Haghigh M Application of Deep Learning in Reservoir Simulation. In: Petroleum Geostatistics 2019, 2019.
23. Tour JM, Kittrell C, Colvin VL (2010) Green carbon as a bridge to renewable energy. Nature materials 9 (11):871-874
24. Leung GCK (2015) Natural Gas as a Clean Fuel. In: Handbook of Clean Energy Systems. pp 1-15. doi:10.1002/9781118991978.hces055
25. Brown SP, Krupnick A, Walls MA (2009) Natural gas: a bridge to a low-carbon future. Issue brief:09-11
26. Kerr RA (2010) Natural Gas From Shale Bursts Onto the Scene. Science 328 (5986):1624. doi:10.1126/science.328.5986.1624
27. Hagoort J (1988) Fundamentals of gas reservoir engineering, vol 23. Elsevier,
28. Iwere FO, Moreno JE, Apaydin OG (2006) Numerical Simulation of Thick, Tight Fluvial Sands. SPE-90630-PA 9 (04):374-381. doi:10.2118/90630-PA
29. Zou C, Zhu R, Liu K, Su L, Bai B, Zhang X, Yuan X, Wang J (2012) Tight gas sandstone reservoirs in China: characteristics and recognition criteria. Journal of Petroleum Science and Engineering 88-89:82-91. doi:https://doi.org/10.1016/j.petrol.2012.02.001
30. Timur A (1968) An investigation of permeability, porosity, & residual water saturation relationships for sandstone reservoirs. The Log Analyst 9 (04)
31. Schön JH (2015) Physical properties of rocks: Fundamentals and principles of petrophysics. Elsevier,

32. Baker RO, Yarranton HW, Jensen J (2015) Practical reservoir engineering and characterisation. Gulf Professional Publishing,
33. Guo B, Ghalambor A (2014) Natural gas engineering handbook. Elsevier,
34. Slider HC (1983) Worldwide practical petroleum reservoir engineering methods. PennWell Books,
35. Hagoort J (1988) Fundamentals of gas reservoir engineering. Elsevier,

6. Modelling Hydraulically Fractured Tight Gas Reservoirs with an AI-based simulator, Deep Net Simulator (DNS)



Statement of Authorship

Title of Paper	Modelling Hydraulically Fractured Tight Gas Reservoirs with an Artificial Intelligence (AI)-based simulator, Deep Net Simulator (DNS)	
Publication Status	☑ Published □ Submitted for Publication	□ Accepted for Publication □ Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	Ghassemzadeh, S. Gonzalez Perdomo, M. Abbasnejad, E. Haghighi, M. (2020) Presented in First EAGE Digitalization Conference and Exhibition and Published in EarthDoc Database	

Principal Author

Name of Principal Author (Candidate)	Shahdad Ghassemzadeh		
Contribution to the Paper	Generated the required data, developed the computer model, analysed the results, and wrote the paper		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	19/11/2020

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Maria Gonzalez Perdomo		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	28/09/2020

Name of Co-Author	Manouchehr Haghighi		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	16/10/2020

Name of Co-Author	Ehsan Abbasnejad		
Contribution to the Paper	Supported with the structure, writing and review of paper, 10%		
Signature		Date	18/11/2020

Please cut and paste additional co-author panels here as required.

TuDIGI07

Modelling Hydraulically Fractured Tight Gas Reservoirs with an Artificial Intelligence (AI)-Based Simulator, Deep Net Simulator (DNS)

S. Ghassemzadeh[1]*, M. Gonzalez Perdomo[1], E. Abbasnejad[2], M. Haghighi[1]

[1] Australian School of Petroleum & Energy Resources (ASPER), University Of Adelaide; [2] Australian Institute for Machine Learning (AIML), University Of Adelaide

Summary

Hydraulic fracturing in tight gas reservoirs increases the connectivity of the well to further areal regions, thus boosting the production as well as the net-present-value of the asset. This type of reservoir typically exhibits considerable uncertainty in rock and fracture properties, which coupled with significant heterogeneity makes history matching, uncertainty quantification, and optimisation time-consuming tasks. Therefore, engineers are always looking for processes to reduce simulation time. Artificial intelligence enables machine-learning to learn from data. This allows for time-consuming fluid flow equations to be explicitly formulated while keeping the accuracy found through the implicit approach. This is achieved through the use of deep learning. In this research, a fully standalone simulator is developed for a range of hydraulically fractured tight gas reservoirs in a 2-dimensional space. Considering the low value of metrics (RMSE<65 psi, MAPE < 0.99%, and $R2 \approx 1$) for training, validation and test sets, the results confirmed that the developed model, Deep Net Simulator (DNS), is accurate and reliable when compared with numerical models. Furthermore, DNS shows remarkable reliability when comparing the results of 140 unseen complete reservoir models over a 4-year period against a numerical simulator. The average value of MAPE for all 140 cases is 10.55%.

Introduction

Reservoir simulation is an area of reservoir engineering in which computer models are used to predict the flow of fluids through porous media and therefore predict the performance of a reservoir under different developmental scenarios. Although the introduction of powerful hardware and software has sped up the reservoir simulation undoubtedly; computational inefficiency and slow running time are still ongoing problems in this area.

Reservoir simulation is typically used to quantify the uncertainties, calibrate the model through history matching and optimise the field production. Each of these tasks requires hundreds, or even thousands of simulation runs to be completed.

These tasks are even more challenging in unconventional reservoirs such as tight gas reservoirs due to their unique geologic characteristics. The most notable feature in these reservoirs is to have permeability values of less than 0.1 md (Law & Curtis, 2002). This attribution makes it impossible to produce at an economical rate in a tight gas reservoir unless the reservoir is hydraulically fractured.

Furthermore, because of the heterogeneity of this complex system, values predicted for the porosity, permeability, thickness and hydraulic fracture properties, which include the permeability, width and length of the fracture, suffer from significant uncertainties. Due to the challenging nature of reservoir characterisation and the stark difference of permeability attributed to fractured and intact zones, it is more time-consuming for a commercial simulator to converge to the final solution when modelling tight gas reservoirs. This makes tasks such as uncertainty quantification, history matching and optimisation significantly lengthy and resource-demanding. Therefore, to increase the efficiency of reservoir management in tight gas reservoirs, seeking a faster alternative is inevitable.

Currently, one of the most popular alternatives is to create a proxy model with the desired output as a replica of a reservoir simulation model. This is done through using machine learning and reduced-order modelling methods (Artus et al., 2014; Knudsen & Foss, 2015; Law & Curtis, 2002; Wang et al., 2019). Another approach is to create a forecasting tool to predict reservoir performance data. The outputs in both these approaches are of the gas flow rate and respective bottom hole pressure. Although these approaches can directly or indirectly speed up the reservoir simulation, they are not independent. In both methods, a numerical simulator must be used either to develop a proxy model of the desired reservoir or to integrate to show the overall performance and pressure distribution of the reservoir.

To overcome this inefficiency, we proposed an alternative approach to numerical simulation (Ghassemzadeh et al., 2019). In the current study, we used deep learning to develop a stand-alone data-driven simulator, as a proxy to the math-based reservoir simulators which is used to predict the pressure of any point across the tight gas reservoir. Unlike the forecasting tool, the developed simulator can be used to predict the reservoir performance in any location in the reservoir. Furthermore, unlike the proxy modelling approach, which is used only for its corresponding reservoir, this model can be used to predict the reservoir behaviour of any reservoir providing that its description and settings lie within the specified range used in this study.

Methodology

In a proxy modelling approach, a dataset is built by running a simulator for slightly different variations of the settings of the corresponding reservoir. However, unlike the conventional proxy models, we propose to create a dataset of reservoirs with varying configurations to allow an intelligent agent to learn the generic characteristics of modelling fluids.

First, we generated a reservoir configuration pool that describes the physical behaviour of the gas flow in tight sand gas reservoirs. In order to create more realistic synthetic reservoir models, these configurations were inspired in respect to real data gathered from reservoirs located in Australia.

Using this configuration pool, we generated N (=50) data files which comprises the necessary reservoir description and settings to model a reservoir with a commercial simulator (ECLIPSE). To create a data file, the required values were randomly chosen from the reservoir configuration pool and then written into a file with the data file format.

Each model is comprised of 40 x 40 x 1 grid cells with one vertical well, which has been hydraulically fractured. We considered that the production well is controlled by the bottom hole pressure (BHP). Therefore, in each scenario, the production well is produced with the allocated well rate, until it reaches the expected BHP. Afterwards, the well rate is reduced to keep the flow according to the BHP. Saturation and relative permeability variables are kept constant as we are simulating single-phase flow. After generating the data file as discussed, ECLIPSE is used to simulate this model. Then, a parsing algorithm was used to break the data file and output of the simulator into a format that can be used to train a deep learning model. Using this method, we generated a training dataset, which consists of around 10,000,000 data points, resulting from the simulation of 50 reservoir models with varied configurations.

To make the deep learning model understand the dynamic concept of the problem, we include time as one of the variables in the model. As it is common in training such models, we normalise the input values with zero mean and unit variance. We utilise a fully-connected deep learning network to model this dataset. The final design has 14 layers with a different number of neurons that varies from 1 to 300. The number of neurons, starting on 20 at the input layer, first increases up to 300 and then decreases to 1 at the output layer, Fig 1. We implement the Adam optimisation algorithm to learn the weights of the neurons. We experiment with different architectures to create the best possible model. We employ the Mean Squared Error (MSE) as the loss to minimise the difference between the estimated cell pressure and the ground-truth from the simulator's. To avoid the overfitting problem, we incorporate dropout regularisation and use the early stopping strategy with a validation set using the mean absolute percentage error (MAPE) as the monitoring metric in the training process. We use Elu and ReLU as the activation functions between layers.

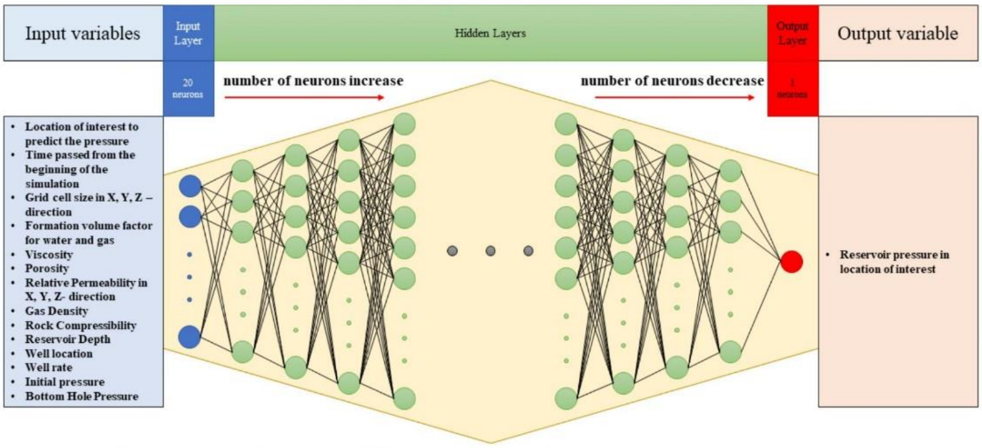

Input variables
Input Layer
Hidden Layers
Output Layer
Output variable
20 neurons
number of neurons increase
number of neurons decrease
1 neurons
• Location of interest to predict the pressure
• Time passed from the beginning of the simulation
• Grid cell size in X, Y, Z – direction
• Formation volume factor for water and gas
• Viscosity
• Porosity
• Relative Permeability in X, Y, Z- direction
• Gas Density
• Rock Compressibility
• Reservoir Depth
• Well location
• Well rate
• Initial pressure
• Bottom Hole Pressure
• Reservoir pressure in location of interest

Figure 1 *Deep learning topology of DNS with 14 dense layers.*

Results and discussion

In this research, we present a data-driven model that can be used to predict the behaviour of a hydraulic-fractured tight sand gas reservoir. The developed model, deep net simulator or DNS, can instantly predict the real-time reservoir pressure during production in any location across the reservoir. Unlike any conventional reservoir simulator, which required the properties of all the grid cells to perform the simulation, DNS prediction relies only on the characteristics of the location (grid cell), distance to the wellbore, initial condition and production settings (Figure 1), without any dependency to the properties of other cells. With this approach, we can simulate any part of the reservoir without the need to simulate the whole reservoir. Using this strategy, we are able to speed up the reservoir simulation significantly, meaning it will only take a fraction of a second to predict the pressure for a specific location. The developed model, deep net simulator or DNS, was trained using our proposed dataset. The MAPE for training, validating and test data are 0.98, 0.99 and 0.99 respectively, the root mean square error (RMSE) for all sets are around 63.24 and the R-squared (R2) in all sets are almost 1, which shows our model achieved impressive performance in modelling such a complex problem. The low values for MAPE, RMSE and with high values for R2 confirm that the developed model has captured the behaviour of the input data accurately. To challenge the data in a complete simulation scenario, we generated 140 unseen case studies. We assigned different values for the fracture length and permeability for the cells connected to the fracture. In addition, completely different initial conditions and production settings were assumed in each case. Table 2 shows the accuracy of DNS in predicting the pressure across the reservoir compared to a commercial simulator. In these cases, fracture length varies from 30 ft to 300 ft, and permeability of the cells connected to the fractures varies from 1000 md to 10000 md.

***Table 2** the accuracy of DNS in predicting the pressure across the reservoir.*

Case	Fracture properties	MAPE (%)	MAE (psi)	MRE (psi)	MSE	RMSE	R2
1	Kf = 2500 md Half Lf = 100 ft	30.56	1.54	-0.01	65155.94	255.26	1.00
2	Kf = 8000 Half Lf = 300 ft	21.34	0.65	-0.01	19023.03	137.92	1.00
3	Kf = 1000 md Half Lf = 300 ft	43.13	1.35	-0.01	11871.67	108.96	0.98
4	Kf = 3000 md Half Lf = 30 ft	61.90	1.83	-0.02	14711.87	121.29	1.00
5	Kf = 10000 md Half Lf = 100 ft	8.80	0.40	0.00	397.29	19.93	0.99

As can be seen in table 2, DNS shows remarkable accuracy in estimating the pressure in these cases. Figure 2 shows the further analysis for the first case mentioned in the table. This figure illustrated the pressure distribution of this case over the drainage area for twelve time steps. As shown in this figure, DNS (left) predicts a similar pattern of pressure drop to the commercial simulator (right). This similarity shows our model can safely be used as a reliable alternative for commercial simulators.

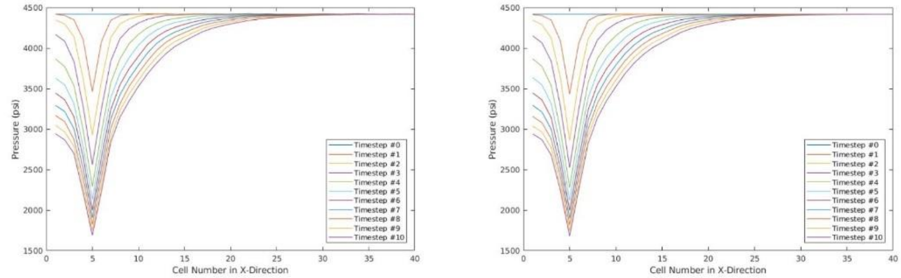


***Figure 2** Pressure distribution throughout the reservoir for different time steps using DNS (left) and a commercial simulator right).*

As previously mentioned, we simulated 140 case studies with the developed model. Figure 3 shows a histogram (left) and cumulative histogram (right) with the accuracy of the DNS for every unseen case study we developed. As shown in this figure, in around thirty percent of simulated cases, DNS shows a remarkable accuracy with MAPE being less than 4%. In addition, over sixty per cent of all case studies have MAPE less than 10%, and ninety per cent of them have MAPE less than 24%. These results confirm that this approach is promising and in the near future, can be considered as a reliable alternative to mathematical-based simulators.

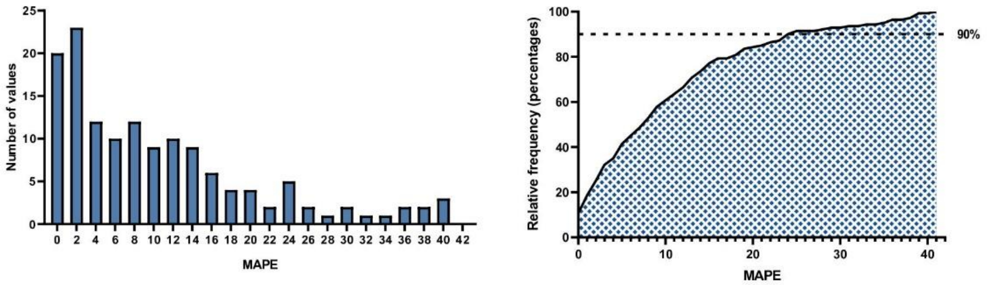


Figure 3 *the histogram of the accuracy of DNS for all the case studies in this study.*

Conclusions

As decision making is the backbone of the business environment, we need to ensure that we are getting accurate and fast responses in order to make a justifiable and profitable decision. These responses are completed through the aid of up to 1000's of simulations, which can take up to days or weeks. This is not fast enough in today's standard, where opportunities can be gone within the day. This is why it has become inevitable to seek a faster alternate approach to reservoir simulation. In this study, deep learning was used to develop a stand-alone data-driven reservoir simulator, DNS, for a hydraulic fracture tight reservoir. The significant contributions of our current research are summarised as follows:

- DNS is capable to accurately estimate the pressure distribution throughout a tight sand gas reservoir, which has been under hydraulic fracture treatment, as long as the reservoir's description and setting lie within the range of training data.
- Unlike a proxy model for a specific reservoir, DNS was developed to cover a wide range of input data.
- The developed model is a fast alternative leading to the exploration of more scenarios in history matching, field management and development planning.
- The MAPE for training, validating and test data are 0.98, 0.99 and 0.99 respectively, the RMSE for all sets are around 63.24, and the R2 in all sets are almost 1.
- The model is validated using 140 case studies, in which the cell's pressure is predicted with an average value of MAPE being 10.55%. Overall, DNS predicts the reservoir behaviour of 60% of cases with MAPE less than 10% and 90% of cases with MAPE less than 24%.

Acknowledgements

The authors are grateful to Alexander Arthurson for his assistance and Santos Ltd for their financial support in this research.

References

Artus, V., Tauzin, E., & Houzé, O. (2014). Efficient Proxies for Numerical Simulation of Unconventional Resources. In SPE/AAPG/SEG Unconventional Resources Technology Conference (pp. 16). Denver, Colorado, USA: Unconventional Resources Technology Conference.

Ghassemzadeh, S., Perdomo, M. G., & Haghigh, M. (2019). Application of Deep Learning in Reservoir Simulation. In Petroleum Geostatistics 2019.

Knudsen, B. R., & Foss, B. (2015). Designing shale-well proxy models for field development and production optimization problems. Journal of Natural Gas Science and Engineering, 27, 504-514.

Law, B. E., & Curtis, J. B. (2002). Introduction to unconventional petroleum systems. AAPG Bulletin, 86, 1851-1852.

Wang, S., Sobecki, N., Ding, D., Wu, Y.-S., & Zhu, L. (2019). Accelerated Compositional Simulation of Tight Oil and Shale Gas Reservoirs Using Proxy Flash Calculation. In SPE Reservoir Simulation Conference (pp. 19). Galveston, Texas, USA: Society of Petroleum Engineers.

7. Conclusion and Future Work

7.1.Concluding Remarks

This study focused on studying the feasibility of the development of an entirely data-driven simulator as an independent alternative to physics-based reservoir simulators. Based on the results of this dissertation, a series of prototype versions of this data-driven simulator were developed for a range of different scenarios. Subsequently, these data-driven simulators were validated via thousands of benchmark cases, consisting of more than 800,000,000 data points.

These developed data-driven models are able to understand and reproduce the physical governing fluid flow for a range of different scenarios: a single-phase oil reservoir in one-dimensional space, a single-phase gas reservoir in two-dimensional space, a single-phase gas reservoir in three-dimensional space, and a hydraulically-fractured tight gas reservoirs in two-dimensional space.

It was shown that the developed simulator, DNS, can predict reservoir behaviour for a wide range of reservoir configurations using the following inputs: time, cell size in the X, Y, Z – direction, formation volume factor for both water and gas, viscosity, porosity, relative permeability in the X, Y, Z- directions, gas density, initial water saturation, rock compressibility, reservoir depth, well location, well rate, initial pressure and bottom hole pressure.

Throughout this dissertation, it is demonstrated that DNS is able to act as an independent clone to numerical reservoir simulators. This simulator could be used to investigate a variety of scenarios during the history matching, uncertainty quantification and field optimisation processes, allowing for faster and better decision-making processes.

The exceptional contributions within this thesis can be summarised as follows:

- This thesis details the logistics of the design and implementation of a data-driven model that is capable of predicting the pressure across the drainage area of a vertical well within a hydrocarbon reservoir. The novelty in this study lies within the implementation of a deep learning network to develop a trained model to act as a comprehensive tool that can act as a fast data-driven replacement for a commercial simulator.
- The results confirm that this approach has the potential to be used in building a pre-trained data-driven simulator that is capable of providing accurate, fast and reliable results for any given reservoir configuration. Unlike conventional approaches to proxy modelling, DNS is not built for any one specific reservoir. Conversely, DNS was built to be used in predicting the reservoir performance for a wide range of reservoir configurations, without a requirement for further training or building processes.
- The simulation was evaluated in terms of precision, speed and reliability. These obtained results were then compared with the physics-based simulator, Schlumberger Eclipse. This resulted in more than 800 varying synthetic reservoirs being generated, each with its own unique description and production settings. Multiple metrics were used in evaluating these case studies; MAE, MRE, MAPE, MSE. RMSE and R2.
- Due to the indirect dependency that DNS holds to the initial and boundary conditions, it can directly simulate specific cells within a particular period. This allows DNS to be incredibly fast compared with traditional physics-based simulators. It results in DNS taking only 1.5e-6 seconds to predict the cell pressure for a single timestep. Through this approach, it takes DNS approximately, on average, 1.18e-4 seconds to predict the pressure trend of a wellbore for five case studies. This means that, on average, DNS is 2e+10 times faster than commercial simulators.

- In this study, the objective of incorporating real field data in the simulation was not achieved. This is due to some limitations associated with DNS in its current form. The current configuration of DNS makes it suitable for single-phase gas reservoirs with one vertical production well, with or without hydraulic fractures, in non-complex geological structures, however, the same approach could be accommodated to model all types of reservoirs with different features and settings. Due to these restrictions, DNS was not evaluated against the real field data, as field data with such attributes was not found. Regardless, the results confirm the feasibility of development of a data-driven simulator. It is believed that DNS can predict the outcome of a commercial simulator, whatever the inputs to this simulator are. This includes synthetic data, semi-synthetic data, and real field data, as long as their attributes lie within the range of the training data.

7.2.Future Work

Future work on this topic should consider the following recommendations:

- To consider multiple wellbores within the reservoirs under study
- To study the effect of multi-phase flow in the accuracy of the data-driven model by including an aquifer and/or incorporating injection-production scenarios within the reservoirs under study
- To consider various reservoir shapes and geometries within the reservoirs under study. This study only focuses on cube-shaped reservoirs.
- To perform history-matching and a field optimisation study to compare the outcomes with a physics-based simulator
- To incorporate the multilayer perceptron (MLP) with recurrent neural networks (RNN) for more complicated scenarios, such as multiphase fluid flow with/without multiple production wells and water/gas injection wells

- To feed the deep learning model with real field data, instead of generating synthetic data using a commercial simulator.

References

Abooali, D. and E. Khamehchi (2019). "New predictive method for estimation of natural gas hydrate formation temperature using genetic programming." Neural Computing and Applications **31**(7): 2485-2494.

Abou-Kassem, J. H., S. Farouq-Ali and M. R. Islam (2013). Petroleum Reservoir Simulations, Elsevier.

Alenezi, F. and S. Mohaghegh (2016). A data-driven smart proxy model for a comprehensive reservoir simulation. 2016 4th Saudi International Conference on Information Technology (Big Data Analysis)(KACSTIT), IEEE.

Alenezi, F. and S. Mohaghegh (2017). Developing a Smart Proxy for the SACROC Water-Flooding Numerical Reservoir Simulation Model. SPE Western Regional Meeting. Bakersfield, California, Society of Petroleum Engineers.

Alenezi, F. and S. Mohaghegh (2017). Developing a smart proxy for the SACROC water-flooding numerical reservoir simulation model. SPE Western Regional Meeting, Society of Petroleum Engineers.

Alipanahi, B., A. Delong, M. T. Weirauch and B. J. Frey (2015). "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning." Nature Biotechnology **33**: 831.

Angermueller, C., T. Pärnamaa, L. Parts and O. Stegle (2016). "Deep learning for computational biology." Molecular Systems Biology **12**(7).

Antonio Cardoso, M. and L. J. Durlofsky (2010). Use of Reduced-Order Modeling Procedures for Production Optimization.

Artun, E. (2017). "Characterizing interwell connectivity in waterflooded reservoirs using data-driven and reduced-physics models: a comparative study." Neural Computing and Applications **28**(7): 1729-1743.

Artun, E., T. Ertekin, R. Watson and M. Al-Wadhahi (2011). "Development of universal proxy models for screening and optimization of cyclic pressure pulsing in naturally fractured reservoirs." Journal of Natural Gas Science and Engineering **3**(6): 667-686.

Artun, E., T. Ertekin, R. Watson, M. J. J. o. N. G. S. Al-Wadhahi and Engineering (2011). "Development of universal proxy models for screening and optimization of cyclic pressure pulsing in naturally fractured reservoirs." **3**(6): 667-686.

Baker, R. O., H. W. Yarranton and J. Jensen (2015). Practical reservoir engineering and characterization, Gulf Professional Publishing.

Barros, P., D. Jirak, C. Weber and S. Wermter (2015). "Multimodal emotional state recognition using sequence-dependent deep hierarchical features." Neural Networks **72**: 140-151.

Beckner, B. L., K. B. Haugen, S. Maliassov, V. Dyadechko and K. D. Wiegand (2015). General Parallel Reservoir Simulation. Abu Dhabi International Petroleum Exhibition and Conference. Abu Dhabi, UAE, Society of Petroleum Engineers.

Brantson, E. T., B. Ju, P. O. Appau, P. H. Akwensi, G. A. Peprah, N. Liu, E. S. Aphu, E. A. Boah, A. A. J. J. o. P. S. Borsah and Engineering (2020). "Development of hybrid low salinity water polymer flooding numerical reservoir simulator and smart proxy model for chemical enhanced oil recovery (CEOR)." **187**: 106751.

Brown, S. P., A. Krupnick and M. A. Walls (2009). "Natural gas: a bridge to a low-carbon future." Issue brief: 09-11.

Chen, B., J. He, X.-H. Wen, W. Chen and A. C. Reynolds (2017). "Uncertainty quantification and value of information assessment using proxies and Markov chain Monte Carlo method for a pilot project." Journal of Petroleum Science and Engineering **157**: 328-339.

Coats, K. J. J. o. P. T. (1969). "Use and misuse of reservoir simulation models." **21**(11): 1,391-391,398.

Cullick, A. S., W. D. Johnson and G. Shi (2006). Improved and more rapid history matching with a nonlinear proxy and global optimization. SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.

Das, A., P. Pradhapan, W. Groenendaal, P. Adiraju, R. T. Rajan, F. Catthoor, S. Schaafsma, J. L. Krichmar, N. Dutt and C. Van Hoof (2018). "Unsupervised heart-rate estimation in wearables with Liquid states and a probabilistic readout." Neural Networks **99**: 134-147.

Deutsch, C. V. and A. G. J. N. Y. Journel (1992). "Geostatistical software library and user's guide." **119**(147).

Eldred, M. E., A. Orangi, A. A. Al-Emadi, A. Ahmad, T. J. O'Reilly and N. Barghouti (2014). Reservoir Simulations in a High Performance Cloud Computing Environment. SPE Intelligent Energy Conference & Exhibition. Utrecht, The Netherlands, Society of Petroleum Engineers.

Ertekin, T., J. H. Abou-Kassem and G. R. King (2001). Basic applied reservoir simulation.

Fanchi, J. R. (2001). Principles of applied reservoir simulation. Boston, Gulf Pub.

Friedmann, F., A. Chawathe and D. Larue (2001). Assessing uncertainty in channelized reservoirs using experimental designs. SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.

Ghasemi, M., Y. Yang, E. Gildin, Y. Efendiev and V. Calo (2015). Fast Multiscale Reservoir Simulations using POD-DEIM Model Reduction. SPE Reservoir Simulation Symposium. Houston, Texas, USA, Society of Petroleum Engineers**:** 18.

Ghassemzadeh, S. and A. H. Charkhi (2016). "Optimization of integrated production system using advanced proxy based models: A new approach." Journal of Natural Gas Science and Engineering **35**: 89-96.

Ghassemzadeh, S., A. H. J. J. o. N. G. S. Charkhi and Engineering (2016). "Optimization of integrated production system using advanced proxy based models: A new approach." **35**: 89-96.

Ghassemzadeh, S., M. G. Perdomo and M. Haghighi (2019). Application of Deep Learning in Reservoir Simulation. Petroleum Geostatistics 2019.

Giusti, A., C. Caccia, D. C. Cireşari, J. Schmidhuber and L. M. Gambardella (2014). A comparison of algorithms and humans for mitosis detection. 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI).

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

Goodfellow, I., Y. Bengio and A. J. D. l. Courville (2016). "Machine learning basics." **1**: 98-164.

Goodwin, N. (2015). Bridging the Gap Between Deterministic and Probabilistic Uncertainty Quantification Using Advanced Proxy Based Methods. SPE Reservoir Simulation Symposium. Houston, Texas, USA, Society of Petroleum Engineers.

Guo, B. and A. Ghalambor (2014). Natural gas engineering handbook, Elsevier.

Hagoort, J. (1988). Fundamentals of gas reservoir engineering, Elsevier.

Hagoort, J. (1988). Fundamentals of gas reservoir engineering, Elsevier.

He, J., J. Xie, X.-H. Wen and W. Chen (2016). "An alternative proxy for history matching using proxy-for-data approach and reduced order modeling." Journal of Petroleum Science and Engineering **146**: 392-399.

Hussain, A., E. Cambria, B. Schuller and N. Howard (2014). "Affective neural networks and cognitive learning systems for big data analysis." Neural Networks **58**: 1-3.

Iwere, F. O., J. E. Moreno and O. G. Apaydin (2006). "Numerical Simulation of Thick, Tight Fluvial Sands." SPE Reservoir Evaluation & Engineering **9**(04): 374-381.

Jansen, J.-D., R. Brouwer and S. G. Douma (2009). Closed loop reservoir management. SPE reservoir simulation symposium, Society of Petroleum Engineers.

Jensen, J. L. (2017). "Comment on “Characterizing interwell connectivity in waterflooded reservoirs using data-driven and reduced-physics models: a comparative study” by E. Artun DOI 10.1007/s00521-015-2152-0." Neural Computing and Applications **28**(7): 1745-1746.

Jeong, D. H. and J. M. Lee (2018). "Enhancement of modifier adaptation scheme via feedforward decision maker using historical disturbance data and deep machine learning." Computers & Chemical Engineering **108**: 31-46.

Journel, A. G. and C. J. Huijbregts (1978). Mining geostatistics, Academic press London.

Kalantari-Dahaghi, A., S. Mohaghegh and S. Esmaili (2015). "Data-driven proxy at hydraulic fracture cluster level: A technique for efficient CO2- enhanced gas recovery and storage assessment in shale reservoir." Journal of Natural Gas Science and Engineering **27**: 515-530.

Kalantari-Dahaghi, A., S. Mohaghegh, S. J. J. o. N. G. S. Esmaili and Engineering (2015). "Coupling numerical simulation and machine learning to model shale gas production at different time resolutions." **25**: 380-392.

Kalantari-Dahaghi, A., S. Mohaghegh, S. J. J. o. N. G. S. Esmaili and Engineering (2015). "Data-driven proxy at hydraulic fracture cluster level: a technique for efficient CO2-enhanced gas recovery and storage assessment in shale reservoir." **27**: 515-530.

Kalantari Dahaghi, A., S. Esmaili and S. D. Mohaghegh (2012). Fast track analysis of shale numerical models. SPE Canadian Unconventional Resources Conference, Society of Petroleum Engineers.

Kaleta, M. P., R. G. Hanea, A. W. Heemink and J.-D. Jansen (2011). "Model-reduced gradient-based history matching." Computational Geosciences **15**(1): 135-153.

Kerr, R. A. (2010). "Natural Gas From Shale Bursts Onto the Scene." Science **328**(5986): 1624.

Killough, J. E. (1995). Ninth SPE Comparative Solution Project: A Reexamination of Black-Oil Simulation. SPE Reservoir Simulation Symposium. San Antonio, Texas, Society of Petroleum Engineers**:** 13.

Kim, H., M. Park, C. W. Kim and D. Shin (2019). "Source localization for hazardous material release in an outdoor chemical plant via a combination of LSTM-RNN and CFD simulation." Computers & Chemical Engineering **125**: 476-489.

Kim, M. and H. Shin (2018). "Development and application of proxy models for predicting the shale barrier size using reservoir parameters and SAGD production data." Journal of Petroleum Science and Engineering **170**: 331-344.

Kulga, B., E. Artun and T. Ertekin (2017). "Development of a data-driven forecasting tool for hydraulically fractured, horizontal wells in tight-gas sands." Computers & Geosciences **103**: 99-110.

Kulga, B., E. Artun and T. Ertekin (2018). "Characterization of tight-gas sand reservoirs from horizontal-well performance data using an inverse neural network." Journal of Natural Gas Science and Engineering **59**: 35-46.

Kulga, B., E. Artun, T. J. C. Ertekin and Geosciences (2017). "Development of a data-driven forecasting tool for hydraulically fractured, horizontal wells in tight-gas sands." **103**: 99-110.

Kulga, B., E. Artun, T. J. J. o. N. G. S. Ertekin and Engineering (2018). "Characterization of tight-gas sand reservoirs from horizontal-well performance data using an inverse neural network." **59**: 35-46.

Leung, G. C. K. (2015). Natural Gas as a Clean Fuel. Handbook of Clean Energy Systems**:** 1-15.

Li, B. and F. Friedmann (2005). Novel multiple resolutions design of experiment/response surface methodology for uncertainty analysis of reservoir simulation forecasts. SPE Reservoir Simulation Symposium, Society of Petroleum Engineers.

Lore, K. G., D. Stoecklein, M. Davies, B. Ganapathysubramanian and S. Sarkar (2018). "A deep learning framework for causal shape transformation." Neural Networks **98**: 305-317.

Mayr, A., G. Klambauer, T. Unterthiner and S. Hochreiter (2016). "DeepTox: Toxicity Prediction using Deep Learning." Frontiers in Environmental Science **3**(80).

Memon, P. Q., S.-P. Yong, W. Pao and P. J. Sean (2014). Surrogate reservoir modeling-prediction of bottom-hole flowing pressure using radial basis neural network. 2014 Science and Information Conference, IEEE.

Mohaghegh, S. D. (2017). Data-Driven Reservoir Modeling, Society of Petroleum Engineers.

Mohaghegh, S. D., R. Gaskari and M. Maysami (2017). Shale Analytics: Making Production and Operational Decisions Based on Facts: A Case Study in Marcellus Shale. SPE Hydraulic Fracturing Technology Conference and Exhibition. The Woodlands, Texas, USA, Society of Petroleum Engineers.

Mohaghegh, S. D., H. H. Hafez, R. Gaskari, M. Haajizadeh and M. Kenawy (2006). Uncertainty analysis of a giant oil field in the middle east using surrogate reservoir model. Abu Dhabi International Petroleum Exhibition and Conference, Society of Petroleum Engineers.

Montgomery, D. C., E. A. Peck and G. G. Vining (2012). Introduction to linear regression analysis, John Wiley & Sons.

Navratil, J., A. King, J. Rios, G. Kollias, R. Torrado and A. J. F. i. B. D. Codas (2019). "Accelerating Physics-Based Simulations Using Neural Network Proxies: An Application in Oil Reservoir Modeling." **2**: 33.

Nwachukwu, A., H. Jeong, M. Pyrcz and L. W. Lake (2018). "Fast evaluation of well placements in heterogeneous reservoir models using machine learning." Journal of Petroleum Science and Engineering **163**: 463-475.

Schmidhuber, J. (2015). "Deep learning in neural networks: an overview." Neural Netw **61**: 85-117.

Schön, J. H. (2015). Physical properties of rocks: Fundamentals and principles of petrophysics, Elsevier.

Shams, M., A. H. El-Banbi and H. Sayyouh (2017). A comparative study of proxy modeling techniques in assisted history matching. SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition, Society of Petroleum Engineers.

Slider, H. C. (1983). Worldwide practical petroleum reservoir engineering methods, PennWell Books.

Stettler, M. and G. Francis (2018). "Using a model of human visual perception to improve deep learning." Neural Networks **104**: 40-49.

Timur, A. (1968). "An investigation of permeability, porosity, & residual water saturation relationships for sandstone reservoirs." The Log Analyst **9**(04).

Tour, J. M., C. Kittrell and V. L. Colvin (2010). "Green carbon as a bridge to renewable energy." Nature materials **9**(11): 871-874.

Wu, H. and J. Zhao (2018). "Deep convolutional neural network model based chemical process fault diagnosis." Computers & Chemical Engineering **115**: 185-197.

Yeten, B., A. Castellini, B. Guyaguler and W. Chen (2005). A comparison study on experimental design and response surface methodologies. SPE Reservoir Simulation Symposium, Society of Petroleum Engineers.

Zheng, J., J. Y. Leung, R. P. Sawatzky and J. M. Alvarez (2019). "An AI-based workflow for estimating shale barrier configurations from SAGD production histories." Neural Computing and Applications **31**(9): 5273-5297.

Zheng, J., J. Y. Leung, R. P. Sawatzky and J. M. J. J. o. E. R. T. Alvarez (2018). "A proxy model for predicting SAGD production from reservoirs containing shale barriers." **140**(12).

Zou, C., R. Zhu, K. Liu, L. Su, B. Bai, X. Zhang, X. Yuan and J. Wang (2012). "Tight gas sandstone reservoirs in China: characteristics and recognition criteria." Journal of Petroleum Science and Engineering **88-89**: 82-91.