APPLICATION OF NON-INTRUSIVE REDUCED ORDER MODELLING ON RESERVOIR SIMULATION USING RADIAL BASIS FUNCTION

A Thesis

by

XIAOMU DONG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Eduardo Gildin
Committee Members,	George Moridis
	Zenon Medina-Cetina
Head of Department,	Jeffery Spath

May 2021

Major Subject: Petroleum Engineering

ABSTRACT

The aim of this thesis is to assess the efficiency of a new data-based method of Non-Intrusive Reduced Order Modeling (NIROM) which can be applied on commercial reservoir simulators and discuss the performance of the NIROM method on reducing the computational running time within an acceptable compromise in accuracy.

NIROM methods are usually constructed by a combination of machine learning techniques and projection-based ROM methods. As opposed to projection-based ROM, such as the proper orthogonal decomposition (POD), NIROM methods treats the non-linear equation as a black box and its approximation is performed by the Reduced-based Function (RBF). In this case, NIROM manifests as a data-driven methodology, whereby the only information the users need is the snapshots of states, e.g., pressures, saturations, temperature, which are output by the high-fidelity model. This means NIROM can be applied to any commercial reservoir simulator.

In this work, two cases are used to investigate the abilities and limitations of the NIROM method. The first case is the UNISIM-I-D three-phase isotropic heterogeneous model without considering heat flow. In this case, the proposed method is applied to obtain an approximation of the pressure and three-phase saturation field within 20 years.

The second case is a 3-dimensional geothermal reservoir with injection and production wells working at the same time. In this model, cold water is injected from the injection wells and hot water is produced from the production wells. Pressure field and temperature field are predicted within 5 years.

In both cases, the NIROM method is proved to be able to significantly reduce the computational time of running the simulation after an offline training and an offline optional validation process

while remain the accuracy of the simulation within certain range. The accuracy of the proxy obtained is case dependent and varies when different timestep interval of the training data set is applied.

DEDICATION

This work is dedicated to my family, my friends and all people who supported me and enlightened me through the two years at TAMU.

ACKNOWLEDGEMENTS

I would like to express my gratitude to:

Dr. Eduardo Gildin, my graduate advisor and the chair of my dissertation committee for his kindness, patience and support in enlightening and guiding me through my research work.

Dr. George Moridis and Dr. Zenon Medina-Cetina, the member of my dissertation committee for their valuable opinions on my work.

All professors I have taken class with at TAMU for their devotion in teaching and research work.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This thesis is supervised by the dissertation committee consisting of Dr. Eduardo Gildin [chair] and Dr. George Moridis [member] of the Harold Vance Department of Petroleum Engineering and Dr. Zenon Medina-Cetina [member] of the Zachry Department of Civil & Environmental Engineering.

Funding sources

This work is supported by a fellowship from the Department of Petroleum Engineering of Texas A&M University and Gildin Faculty Fellowship.

NOMENCLATURE

B	Radial basis function matrix
c	Controls
f	Surrogate model
n_g	Total number of grid blocks
n_t	Number of timesteps of the simulation for each set of controls
o_n	POD order of the state variable n
p	Pressure
$r_i^{(c)}$	Reduced snapshot of the state at timestep i with the reservoir control c
R	Whole reduced snapshots matrix
R_n	Reduced snapshots matrix for the state variable n
s_i	Snapshot of the state at timestep i
So	Oil saturation
Sw	Water saturation
Sg	Gas saturation
T	Temperature
u	Number of sets of controls
W	Weighting factor matrix
w_{k_m}	Weighting factor applied on each distance
x_n	Snapshot of the state variable n
X_n	Snapshots-matrix of the state variable n

Y	Label value matrix
β	Radial basis function
γ	Shape factor
ϕ	Whole POD projector
ϕ_n	POD projector for the state variable n

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Reservoir simulation is an essential and yet time-consuming process in reservoir engineering. Thousands or even more runs of reservoir simulation are commonly needed in optimization or history matching problems. This can already cause computational issues even some assumptions aiming at simplifying the problems like isothermal reservoir are made. However, with the recent focus on carbon free and sustainable production, geothermal reservoirs are becoming a key component on this transition to cleaner energy. In this case, temperature becomes an additional state variable and even more computational resource needs to be input when doing reservoir simulation. Therefore, when solving such problems with traditional method of implicit pressure, explicit saturation (IMPES) or fully implicit method, expensive high-performance computers are usually needed.

Reduced order modelling (ROM) is developed as a possible solution to increase the efficiency of reservoir simulation, especially if multiple calls of the high-fidelity model is needed as in the case of optimization and history matching. ROM can also be used as complementary to high performance computing devices to lower even further the computational cost associated with very large-scale simulations. ROM aims at reducing the computational time of running simulations with an acceptable compromise in accuracy which is commonly used in engineering problems related to dynamical systems. The idea of it is to reduce the dimension of state space or the degrees of freedom with multifarious algorithms and then simulate with a smaller model.

To apply ROM on reservoir simulation, many algorithms have been developed and explored in the recent years. One commonly used method that many of those algorithms are based on is proper orthogonal decomposition (POD)[1]. The main idea of POD is to project a high dimensional state variable matrix onto a new system of coordinates in which the dimension of the original matrix is

reduced to a much lower order but at the same time preserve the original information as much as possible by keeping the largest principle-values. The most popular algorithm based on POD which is frequently used in reservoir simulation is POD-Galerkin method[2] and its variants. During the training process of POD-Galerkin method, the reduced basis is computed from the training snapshots and it is then used in the online stage to construct the reduced matrix equation of state variables. The reduced equation is solved in the reduced subspace to obtain the next reduced state, and the next reduced state will be project back to the original space after each iteration. The computational time of projecting the reduced state back to the original space for each iteration offsets some of the time reduced by solving the non-linear equation in the reduced subspace. Some subsequent methods are developed to solve this issue. Discrete empirical interpolation method (DEIM)[3] is one of the algorithms that has successfully improved the low efficiency of projecting the reduced state iteratively. DEIM is combined with several other ROM methods to further reduce the computational time and improve the accuracy. Trajectory-based DEIM (TDEIM)[4] is the combination of trajectory piecewise linearization (TPWL)[5] and DEIM. In this method, the new states are represented as the linear combination of the previous trajectory simulated by the high-fidelity model.

All the ROM methods introduced above can be categorized as intrusive ROM methods. It means that the derivation converting the non-linear equation to lower dimensions is required which is not feasible in all the scenarios using analytical methods. Also, to successfully implement these algorithms, the access to the source code of the high-fidelity simulator is necessary which may cause unavoidable confidential issues. To overcome these problems, several non-intrusive ROM (NIROM) methods are developed which are usually the combination of machine learning techniques with ROM methods. When applying NIROM methods, the non-linear equation is

treated as a black box[6] and the only thing the users need is the snapshots of states which are output by the high-fidelity model in several different control groups. This means NIROM can be potentially applied to any commercial simulators.

Along with the rapid development in machine learning techniques in recent years, NIROM methods are being more actively explored, and as a result, have significant improvements in both the accuracy and the efficiency of the method. Dynamic mode decomposition (DMD)[7] and its variants are an example of these NIROM methods which have been applied to solve several engineering problems especially in fluid dynamics and have had good performance in accuracy and efficiency. The idea of DMD method is to find a matrix to linear transform the current snapshots-matrix to the snapshots-matrix one step forward in the training process. In the online stage, the group of transform matrices will be multiplied to the snapshots-matrix iteratively to obtain the predicted state one by one. However, this linear relationship may not be enough to capture the nature of all the dynamical systems. Even so, DMD has provided a good framework for the following researchers to make modifications on. One method that can be used as an alternative of the assumption of linear relationship is radial basis function method (RBF)[8 9 10]. RBF is a method based on distance between the training data points which has increased the non-linearity of the model. The stability of the method of combining RBF with NIROM framework fully depends on the stability of the solution of the high-fidelity model, which avoids the issue that normal ROM methods are sometimes unstable[11]. RBF has been widely applied for predicting the PVT properties of the oil in the reservoir[12]. However, it has not been applied to predict full state field of the reservoir. In this work, a new NIROM method which based on POD and RBF will be introduced and discussed.

1.1. Primary objectives

The primary objectives of this thesis are to:

(1) Investigate the ability of the new NIROM method in reducing computational time of reservoir simulation under isothermal and non-isothermal conditions.

(2) Evaluate the accuracy of the new NIROM method in predicting the values of state variables (i.e., pressure, saturations, temperature) of reservoir models.

1.2. Thesis scope

In order to achieve the two objectives proposed, we will:

(1) Show the implementation workflow of the new NIROM method

(2) Present two case studies to show the application of the NIROM method on the UNISIM-I-D reservoir model and the simple geothermal model to evaluate the abilities of NIROM to predict pressure field, saturation field, temperature field with different well flowrate setups. The models used are shown underneath.

1.2.1. UNISIM-I-D[13]

In the first case study of this thesis, UNISIM-I-D model is used which is a 3D isotropic heterogeneous reservoir model with one fault and four production wells (not fully perforated). The dimensions of this model are 58 * 81 * 20 (93960 grids) and the number of active grids is 36403. The mean porosity is 0.1434 and the standard deviation is 10.83. In this thesis, three-phase reservoir simulations are run on this model to obtain the training sets for the training of the proxy and the testing sets to evaluate the performance of the proxy. The model and the well locations are shown in **Figure 1.1**.

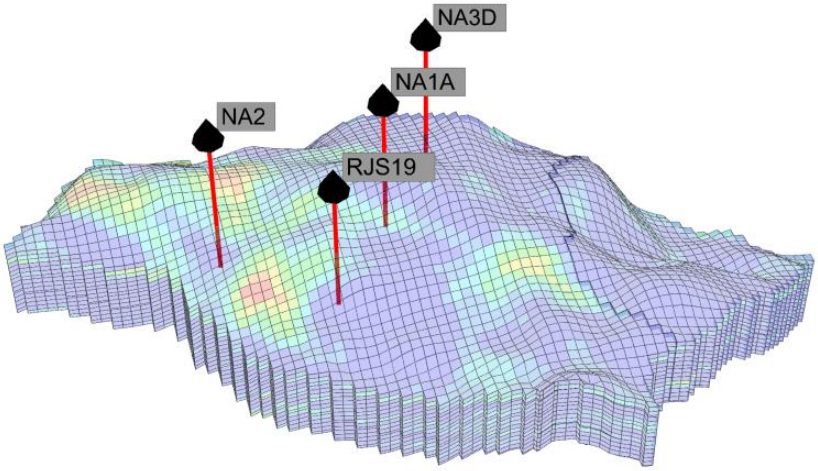


Figure 1.1 UNISIM-I-D and well locations.

1.2.2. Geothermal model

Another case study is a relatively simple geothermal 3D isotropic, homogeneous model with 8 injection wells and 7 production wells. In this case study, the proxy of the high-fidelity geothermal model is used to predict the pressure and temperature field. The dimensions of this model are 80 * 80 * 15 (96000 grids). In this case, the injection wells and production wells are perforated from layer 3 to layer 12 and are opened at the same time. The initial reservoir temperature is 170 ℃ and the injection wells are injecting cool water at 25 ℃. The initial reservoir pressure is 25000 kpa. The dimensions and the locations of the injectors and producers are shown in **Figure 1.2** underneath.

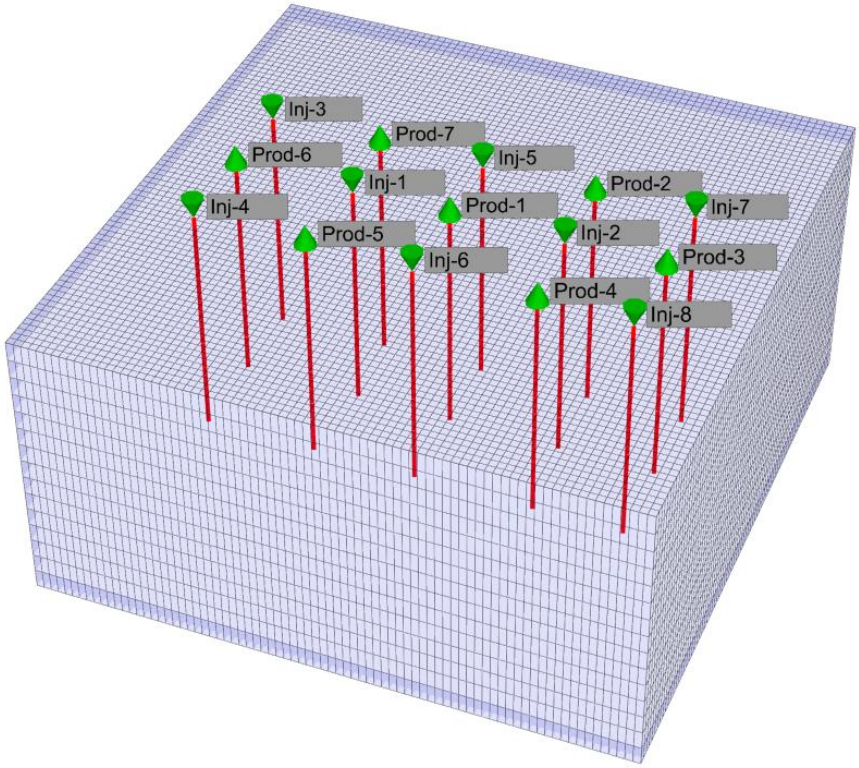


Figure 1.2 Well locations of 8 injection wells and 7 production wells of the 3-dimentional geothermal model

2. METHODOLOGY

In reservoir simulation, the traditional method of computing the variation of states of two continuous timesteps is to solve an equation which contains a high dimensional Jacobian matrix in an iterative pattern[14]. Although many new types of linear solver have been invented and optimized in order to enhance the performance of traditional reservoir simulation, the high computational time is still a big issue especially when many runs of simulation are needed in problems like optimization and history matching.

Under this background, ROM methods are applied on reservoir simulation to reduce the computational time while preserve most of the accuracy. As contrast of the idea of enhancing the efficiency of the linear solver, ROM methods take a different strategy which is reducing the computation required through reducing the size of the model. One commonly used method to achieve this is to simplify the matrix computation by conducting it in a lower dimensional space.

POD is a good tool for converting a higher dimensional matrix to a lower dimension while preserve the most important information of principle components in the original matrix. It is developed from Principal component analysis (PCA) which was firstly invented by Karl Pearson[15]. Karl Pearson thought a multi-dimensional ellipsoid can be used to fit the dataset and the principal component can be defined as the axes of the multi-dimensional ellipsoid. If the axis is short, it means there cannot be much variance along the axis and the amount of information contained by its corresponding principal component is small. Thus, if we ignore the shortest few principle components, not too much information will be lost. POD is based on the theory of singular value decomposition (SVD). SVD was discovered by several mathematicians independently and Carl Eckart and Gale J. Young first proved SVD for general rectangular and complex matrices[16] which has guaranteed the stability of POD.

POD has enhanced the efficiency of computation of high dimensional matrices. However, the computational time reduced by solving linear equations in lower dimensional space is offset by the time spent on projecting the reduced state back to real space in each iteration. Also, modification to the source code of the simulator is not feasible under all circumstance, especially to commercial simulators due to confidential issue. Therefore, the technique of NIROM was developed where a proxy[17 18] is used as a replacement for the iteration process of solving linear equations. The proxy is usually trained by machine learning techniques, and as contrast of expressing the physics relations by partial differential equations, the proxy learns patterns of the evolvement of the system directly from the training data. Then, the pattern learnt will be used for the purpose of prediction. The training data are usually several sets of continuous snapshots from several similar control setups of the reservoir[19]. Snapshots of state are a group of column vectors in which all the values of state variables in a model are stacked according to time. They are the most direct representation of the evolvement of the state of a system and are commonly used in data-based ROM problems related to dynamic system.

There are many methods that have been used to build proxy for different parts of reservoir simulation using machine learning techniques[20 21]. In this work, the algorithm used for predicting the evolvement of state is RBF based on the method of Xiao, D. et al[11]. In 2000, Buhmann applied RBF as a tool for approximate numerical solutions of elliptic partial differential equations[8]. In 2006, Schaback and Webdkabd showed the applications of RBF in machine learning and meshless methods for solving partial differential equations. In their opinion, radial basis function (kernel) can be viewed as bell-shaped functions like Gaussians. They can be shifted, scaled, and superimposed with weights to form new functions[9]. Now, the sum of radial basis functions from

the training dataset points is commonly used for interpolating the multi-dimension surfaces constructed by the points in many engineering fields.

In this section, we introduce the workflow of the new NIROM method and the several moving parts and algorithms that are used to construct a proxy for a high-fidelity reservoir simulation model. The workflow of the NIROM method used here can be divided into 5 steps and the flow chart is shown in **Figure 2.1**.

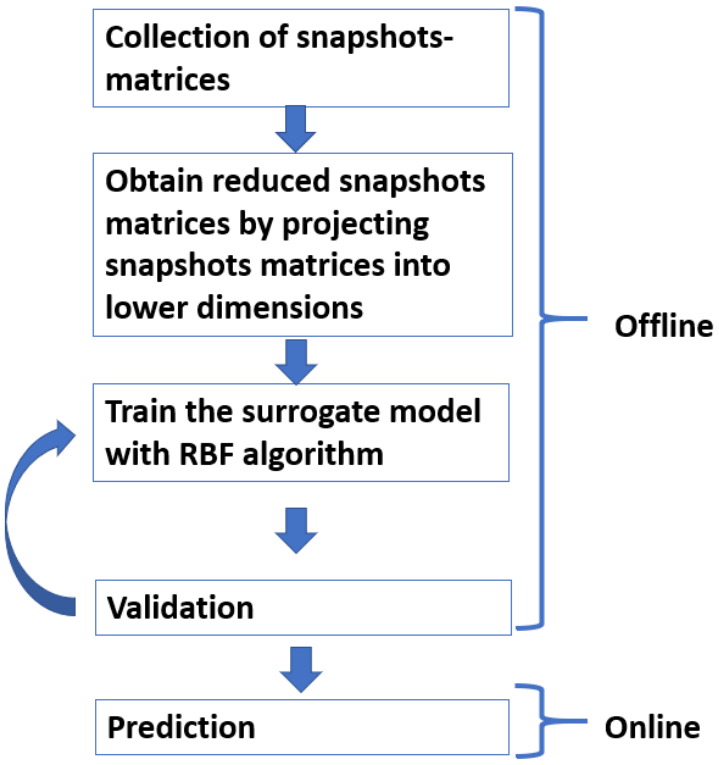


Figure 2.1 Work flow of the new NIROM method

Firstly, high-fidelity simulations are performed to collect the snapshots-matrices from different controls settings (inputs). The schematic illustration of snapshots of any state variable is shown in **Figure 2.2**. These matrices are the representation of the evolvement of the dynamic system. Our goal is to learn the pattern of the evolvement from the snapshots-matrices, and then, the pattern

can be used to predict the variation of new set of input. To achieve this, training data sets need to be built from the snapshots-matrices collected. However, the dimensions of the snapshots-matrices are usually very high especially the row dimension, because the row dimension is equal to the number of grid blocks used in the model. If use the snapshots-matrices directly for training, it will be very computational inefficient that the time cost may be even more than running the full-size simulation.

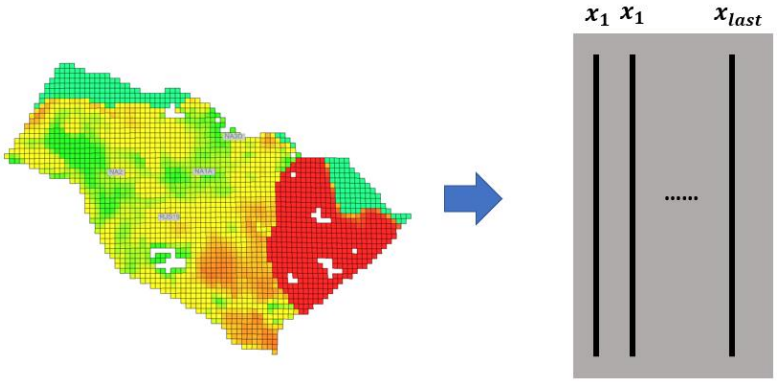


Figure 2.2 Schematic illustration of snapshots of any state variable.

Therefore, the second step is to implement POD to the snapshots-matrices to determine a low dimensional basis set. As introduced above, POD can significantly reduce the row dimension of a matrix by projecting it onto a projector (the row dimension of the matrix can be reduced to few tens according to the setup of the user). The efficiency of training can thus be significantly improved when using reduced snapshots matrices compared with using full snapshots matrices.

Thirdly, the reduced snapshots-matrices will go through the training process to obtain the surrogate model. In the training process, each reduced snapshot in the matrices will act as a training data point, and the distances between each two of these data points will be computed. The value of the

radial basis function of the distances are used as a criteria to quantify the influence between each two data points (the larger the value of the distance, the less influence of one data point to the other data point). Then, the weighting factor assigned on each training data point is computed. When we have new data points (testing data), The evolvement on the new input is predicted by computing the weighted sum of the radial basis function of the distance between the new data point to all training data points.

Fourthly, an algorithm that aims at improving the performance of the surrogate model to capture multiple trends of the variation of the state is implemented. In this algorithm, a validation set (reduced snapshots-matrix) is needed, and a validation process is performed. In this step, a group of surrogate models are collected, and they are combined as the final proxy of the high-fidelity model. This step is optional and depends on the complexity of the model simulated. The first four steps are all performed offline. Finally, the proxy (non-intrusive model) is used to predict the fields of states in the test case. The details of each step are explained underneath.

2.1. Step 1: Collection of snapshots-matrices

As mentioned above, snapshots of a system are the most representative information about the pattern of the evolvement of the system. The proxy will directly learn from the snapshots through training. Therefore, before the training process, snapshots of the states (e.g., pressure, saturations, temperature) of the reservoir under similar control setups need to be collected in advance. The snapshots-matrix of one state variable of one set of simulation control is a two-dimensional matrix with n_g rows and n_t columns where n_g denotes the total number of grid blocks and n_t denotes the number of timesteps of the simulation. For training, several snapshots-matrices from different simulation controls need to be taken. All these snapshots-matrices will be concatenated in a row

and thus the dimensions of snapshots-matrix for one state variable are (n_g, n_t * u), where u denotes the number of controls.

Take pressure as an example, if the user decides to use 5 different sets of controls for training, the number of grid blocks is 10000, and the number of the total timesteps is 100, each snapshot of pressure $x_{p_{i,j}}$, $i \in [1,\ u]$, $j \in [1,\ n_t]$, then the final snapshots matrix of pressure X_p will look like the matrix shown in **Figure 2.3**. The buildup of snapshots-matrix for other state variables are following exact the same method as pressure.

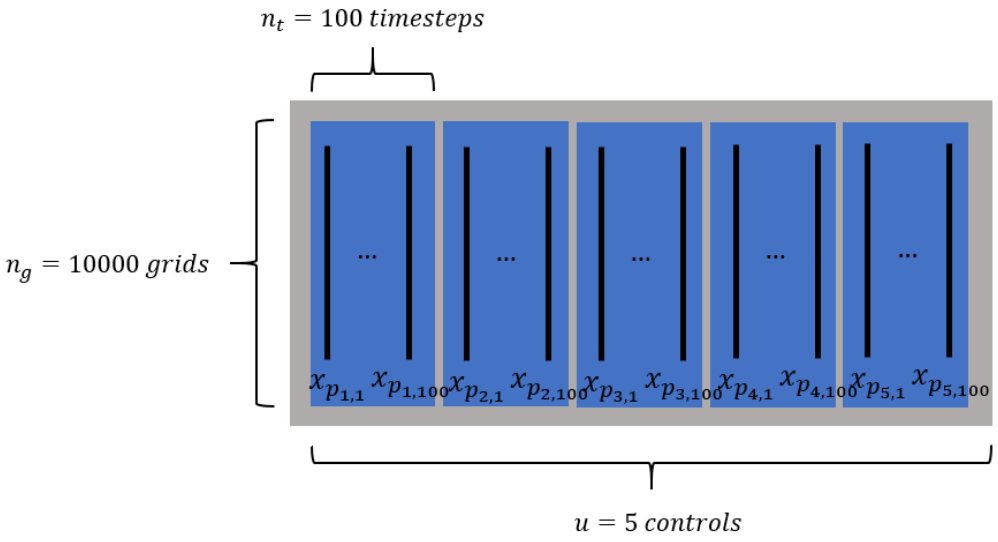


Figure 2.3 Schematic illustration of building up the snapshots-matrix of pressure

2.2. Step 2: POD of the training set

The row dimension of snapshots-matrices is high because it is equal to the number of grids used in the model. Thus, if the snapshots-matrices are directly used for training, the efficiency of computation will be very low. Therefore, in step 2, POD is implemented to the training set collected in step 1 following the procedure shown in **Figure 2.4** to reduce the row dimension of

the training data. The reduced snapshots matrix R and the POD projector ϕ obtained from POD will be recorded. R will be used for training, and ϕ will be used for projecting the reduced snapshots into real space.

POD of the training set:

Data: $X_p,\ X_{So},\ X_{Sw},\ X_{Sg},\ X_T$

$o_{p,}\ o_{So,}\ o_{Sw,}\ o_{Sg,}\ o_T$

Result: R, ϕ

1 for n in $[p, So, Sw, Sg, T]$:

2 $[U, S, V] = SVD(X_n)$

3 $\phi_n = U(:, 1: o_n)$

4 ${R_n} = {\phi_n}^t * X_n$

5 end

6 $R = [R_p; R_{So}; R_{Sw}; R_{Sg}; R_T]$

7 $\phi = diag(\phi_p, \phi_{So}, \phi_{Sw}, \phi_{Sg}, \phi_T)$

8 return R, ϕ

Figure 2.4 Algorithm of implementing POD on the training set

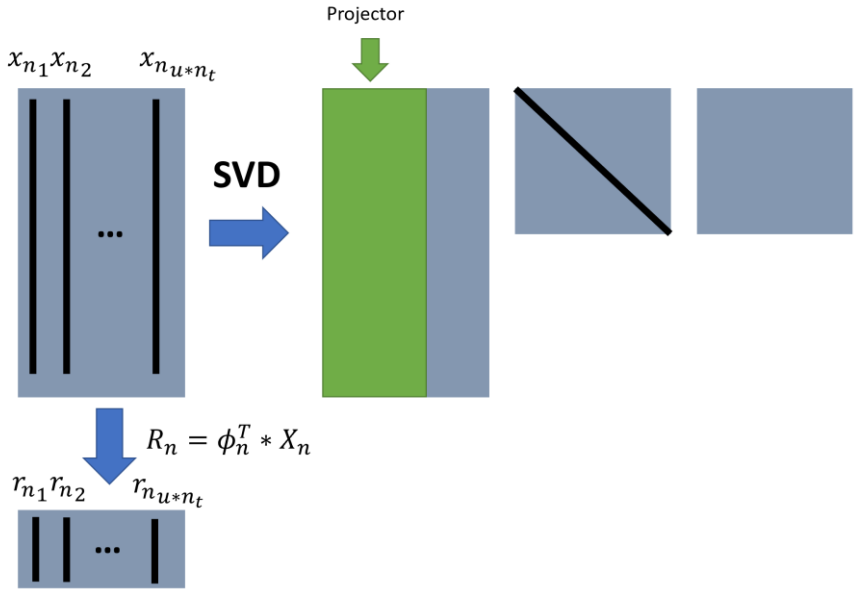


Figure 2.5 Schematic illustration of applying POD on the snapshots-matrix of one state variable.

The schematic illustration of POD is shown in **Figure 2.5**. The main idea of POD is to use Singular Value Decomposition (SVD) to express the snapshots-matrix as the product of three decomposed matrices, and then define a projector from the first matrix to apply on the snapshots-matrix itself. After SVD, first o_n columns of the first decomposed matrix are taken as the projector applied on the full snapshots. o_n denotes the order of POD. When the value of o_n is large, it means more information regards to the principle components in the original space is captured in the reduced snapshots after projection, and thus the prediction results of the NIROM method using the reduced snapshots for training is expected to be more accurate. However, longer computational time in both the offline and online process is needed for larger o_n value because of computation between higher dimensional matrices. Therefore, the balance between computational time and the accuracy of the model needs to be considered. The method to verify whether enough information is captured in the reduced snapshots is to compute the energy percentage of the principle components contained by the reduced snapshots. The energy percentage is computed by using the summation of the singular values selected divided by the summation of all the singular values. When the energy percentage is over 98%, we can say the POD order is high enough.

After obtaining the projector, we can compute the reduced snapshots matrix for each state variable using **Eq.**2.1.

$$R_n = {\phi_n}^t * X_n \quad n \in [p, So, Sw, Sg, T] \tag{2.1}$$

After collecting R_n and ϕ_n, R_n are concatenated on one column to form R and ϕ_n are aligned on the diagonal of the matrix ϕ.

2.3. Step 3: NIROM training using Radial Basis Function (RBF)

As mentioned before, a surrogate model f is used to represent the variation between two continuous timesteps instead of the solution of a linear equation by iteration in traditional reservoir simulator. As shown in **Eq.** 2.2. $r_i^{(c)}$ denotes the reduced snapshot of states at timestep i with the reservoir control c, and the evolvement from timestep i to timestep $i+1$ is estimated by the value of f.

$$r_{i+1}^{(c)} = r_i^{(c)} + f\left(r_i^{(c)}, c\right) \qquad 2.2$$

If we use a term $y_i^{(c)}$to represent $r_{i+1}^{(c)} - r_i^{(c)}$, then we have obtained all the elements for training f. $r_i^{(c)}$is the training data and $y_i^{(c)}$is the corresponding label. To obtain f, RBF is applied on R collected in step 2 which contains all the reduced snapshots $r_i^{(c)}$ to obtain W, Δ, D_p which are the weighting factor matrix, the scaling factor and the scaling factor diagonal matrix needed in the expression of f. The algorithm is shown in **Figure 2.6**.

Training with RBF:

Data: R, c
Result: W, Δ, D_p

$o = \text{sum}(o_{p,}\ o_{So,}\ o_{Sw,}\ o_{Sg,}\ o_T)$
$r\text{eshape } R \text{ to } (o, n_t, u)$
$x = R(:,1{:}\,end-1,:)$
$y = R(:,2{:}\,end,:) - R(:,1{:}\,end-1,:)$
$r\text{eshape } x, y \text{ to } (o, (n_t-1)*u)$
$W = zeros(o, (n_t-1)*u)$
$B = zeros((n_t-1)*u, (n_t-1)*u)$
for i = 1: $(n_t-1)*u$
    for j = 1: $(n_t-1)*u$
        compute δ with eq. 8
        compute D_p with eq. 7
        compute Δ with eq. 6
        d = compute distance between $x(i,:)$ and $x(j,:)$ with eq. 5
        $B(i,j) = \text{compute } \beta(d)$ with eq. 4
    end
end
for k = 1: o
    $W(k,:) = B^{-1} * y(:,k)$
end
return W, Δ, D_p

Figure 2.6 Algorithm of obtaining W, Δ, D_p used for the expression of surrogate model f.

RBF algorithm can be expressed by **Eq.** 2.3, and **Eq.** 2.4 to **Eq.** 2.6 are the extensions to the terms in **Eq.** 2.3. In **Eq.** 2.3, d (•, •) denotes the distance between two training data points, w_{k_m} is the weighting factor applied on radial basis function of each distance, which is decided by the training process, β is the radial basis function selected by the user.

As mentioned before, radial basis function is a bell shape function of the distance between two data points. The bell shape implies that when the two data points are close to each other (the distance is close to zero), the value of radial basis function reaches to its maximum, which means the two data points are closely related to each other. The aim of the RBF algorithm is to solve the

weighting factors which are put on the radial basis function of each distance. If we write **Eq.** 2.3 in the matrix format as shown in **Eq.** 2.9, and then it can be shortened as $B * w = Y$, and the process of solving the weighting factors is simply solving a linear equation. As contrast of solving linear equations iteratively in traditional reservoir simulator, the weighting factors only need to be solved once for each POD order.

$$f_k(r_i, c_j) = \sum_{m=1}^{u*(n_t-1)} \sum_{n=1}^{u} w_{k_m} * \beta\left[d(r_i, c_j), (r_m, c_n)\right] = y_k(r_i, c_j) \quad 2.3$$

where

$$i = 1, 2 \dots \dots u * (n_t - 1)$$

$$j = 1, 2 \dots \dots u$$

$$k = 1, 2 \dots \dots o$$

Eq. 2.4 shows the radial basis function selected in this work, where d denotes the distance and γ is a shape factor used for adjusting the shape of radial basis function.

$$\beta(d) = \frac{1}{\sqrt{d^2 + \gamma^2}} \quad 2.4$$

Eq. 2.5 shows the definition of distance between two data points, where Δ is a scaling factor defined by **Eq.** 2.6, and D_p is a diagonal scaling factor matrix which is defined by **Eq.** 2.7 and the scaling factors on the diagonal are defined by **Eq.** 2.8**.**

$$d^2\left((x_1, c_1), (x_2, c_2)\right) = \frac{||x_2 - x_1||^2}{\Delta^2} + c^T D_p c \quad 2.5$$

$$\Delta = \frac{1}{u(n_t - 1)} \sum_{n=1}^{n_t-1} \sum_{k=1}^{u} ||r_{n+1}{}^{(c_k)} - r_n{}^{(c_k)}|| \quad 2.6$$

$$D_p = \begin{bmatrix} \frac{1}{\delta_1^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\delta_{u-1}^2} \end{bmatrix} \tag{2.7}$$

$$\delta = mean\big(c(:,1:u-1) - c(:,2:u)\big) \tag{2.8}$$

$$\begin{bmatrix} \beta\{d((r_1;c_1),(r_1;c_1))\} & \dots & \beta\left\{d\left((r_1;c_1),(r_{(n_t-1)*u};c_u)\right)\right\} \\ \beta\{d((r_2;c_1),(r_1;c_1))\} & \dots & \beta\left\{d\left((r_2;c_1),(r_{(n_t-1)*u};c_u)\right)\right\} \\ \dots & \dots & \dots \\ \beta\left\{d\left((r_{(n_t-1)*u};c_u),(r_1;c_1)\right)\right\} & \dots & \beta\left\{d\left((r_{(n_t-1)*u};c_u),(r_{(n_t-1)*u};c_u)\right)\right\} \end{bmatrix}$$

$$\begin{bmatrix} w_{k_1} \\ w_{k_2} \\ \dots \\ w_{k_{u*(n_t-1)}} \end{bmatrix} = \begin{bmatrix} y_{k_1} \\ y_{k_2} \\ \dots \\ y_{k_{u*(n_t-1)}} \end{bmatrix} \tag{2.9}$$

Now that B matrix and Y vector is known, w can be easily solved by $w = B^{-1}Y$. When k varies from 1 to o, o sets of such equation can be written and solved, which is shown in **Figure 2.7**.

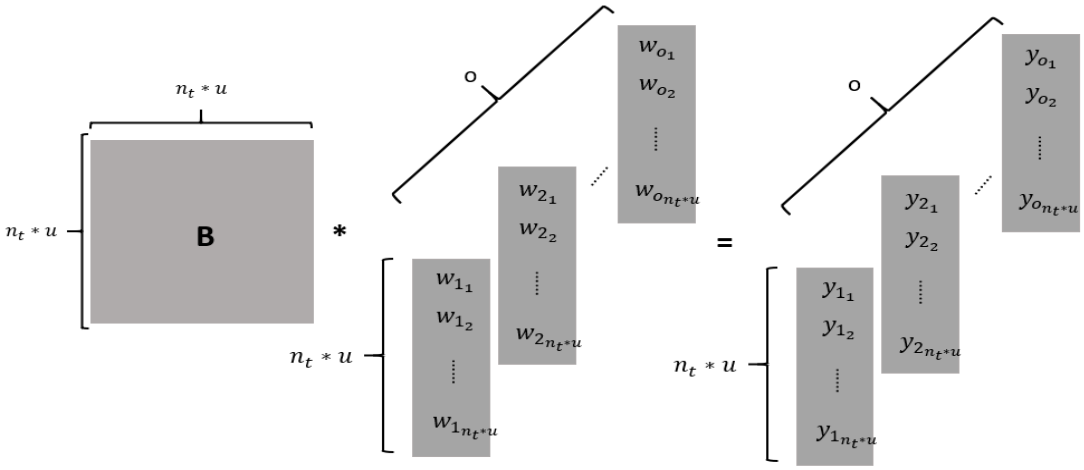


Figure 2.7 Schematic illustration of solving the weighting factors which will be used in the surrogate model for predicting the value of the state variables in each grid block.

2.4. Step 4: Improve the performance of the surrogate model in capturing multiple trends

The variation of the state of a system usually does not follow a smooth trend from beginning to the end, which means there may exist sharp turning points when the variation of state switches from one trend to another. For example, in **Figure 2.8**, the variation of oil saturation can be easily separated into 3 pieces. The oil saturation of the grid holds constant for a while when the saturation front has not reached to the grid. Then, it decreases sharply after the oil starts to be drained from the grid. Finally, when the oil saturation reduces to the residue oil saturation, it holds constant again. It is harder for the surrogate model to predict the states at such turning points because of the high non-linearity at the turning points. Only one surrogate model is unable to capture both the general variation trend of the state and the switch in multiple trends. Therefore, an algorithm is used to recognize the turning points automatically and then improve the performance of the surrogate model to capture multiple trends. The algorithm is shown in **Figure 2.9**.

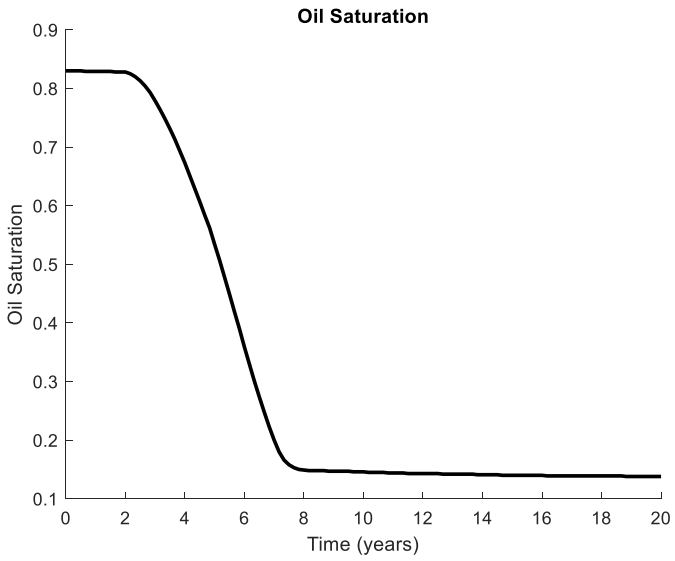


Figure 2.8 Example of multiple trends of variation of the state.

Identify the turning points and obtain the proxy of the high-fidelity model:

Data: $W, \Delta, D_p, s_1^{(c_v)}, R, \phi$

Result: $t_{turn}, W(i), \Delta(i), D_p(i)$, where $i \in [0, length(t_{turn})]$

/* Predict the states on the validation set */
Obtain S with the algorithm shown in Figure 2.10
/* Compute the average relative error of the state variables, the
derivative and the product of left and right derivative. The dimension of
$relErr$, $derRelErr$, $prodLR$ are (1, $n_t + 1$) */
$relErr = compute\ relative\ error\ (S, S_h)$
$derRelErr = compute\ deriative\ of\ relative\ error(\ relE)$
$prodLR = compute\ product\ (derRelErr)$
/* Identify the turning points */
for $prod$ in $prodLR$
    if $prod < threshold$
        append the corresponding time of $prod$ in t_{turn}
    end
end
/* Train the surrogate model after each turning point **/**
for i = 1:$length(t_{turn})$
    $t = t_{turn}(i)$
    $X_n = take\ the\ snapshots\ after\ t\ in\ X_n$
    $R(i), \phi(i) =$ Implement POD to X_n following Figure 2.4
    $W(i), \Delta(i), D_p(i) =$ Training the new surrogate model following Figure 2.6
end
$W(0) = W, \Delta(0) = \Delta, D_p(0) = D_p, \phi(0) = \phi$
return $W(i), \Delta(i), D_p(i)$, where $i \in [0, length(t_{turn})]$

Figure 2.9 The algorithm of identifying the turning point of different trends of variation of the state. The method of forming the proxy of the high-fidelity reservoir simulation model which is the combination of a group of surrogate models with the parameter matrices $W(i), \Delta(i), D_p(i)$, where $i \in [0, length(t_{turn})]$.

In this algorithm, a validation set is collected from the high-fidelity model in the same way as collecting the training set. The initial condition of the validation set s_1 is used as the input, and the states after are predicted by the surrogate model trained in step 3. The algorithm of the prediction process is shown in **Figure 2.10**. Taking another look at the **Eq.** 2.2, f can now be expressed by

W, Δ, D_p obtained in step 3 and the user can input arbitrary controls c into f. The reduced snapshot of the state at the beginning of the simulation r_1 can be computed by projecting s_1 on ϕ using $r_1 = \phi^T * s_1$ where s_1 is the first state. So far, all the reduced snapshots of the states can be computed iteratively. To compute the real value of the states, we only need to project the reduced snapshots back to real space by **Eq.** 2.10.

$$snapshots\ predicted\ by\ NIROM = reduced\ snapshots\ computed\ by\ NIROM * (\phi^T)^{-1} \quad 2.10$$

Prediction on the validation set:

Data: $W, \Delta, D_p, s_1, R, \phi$
Result: S

$R = R(:, 1: end - 1, :)$
$r_1 = \phi^T * s_1$
$rCurrent = r_1$
$redS = zeros(o, n_t + 1)$
$redS(:, 1) = rCurrent$
for k = 1: n_t
    for i = 1: o
        for j = 1: $(n_t - 1) * u$
            d = compute distance between $x(i, :)$ and $x(j, :)$ with eq. 5
            $f(i, 1) = f(i, 1) + W(i, j) * \beta(d)$
        end
    end
    $rNext = rCurrent + f$
    $redS(:, k + 1) = rNext$
    $rCurrent = rNext$
end
$S = (\phi^t)^{-1} * redS$
return S

Figure 2.10 The algorithm of predicting the snapshots-matrix S with the surrogate model parameterized by W, Δ, D_p and the initial condition s_1 as input.

After predicting the states with the initial condition of the validation set, the predicted result is compared with the result collected from the high-fidelity model, and the average relative error of

all state variables is computed for each timestep. The NIROM method usually has the worst performance near to the sharp turning points of the state, and thus the average relative error usually reaches to the local highest point when the variation trend of state switches. By identifying these local highest points in average relative error of all state variables, the possible turning points of the state is thus recognized.

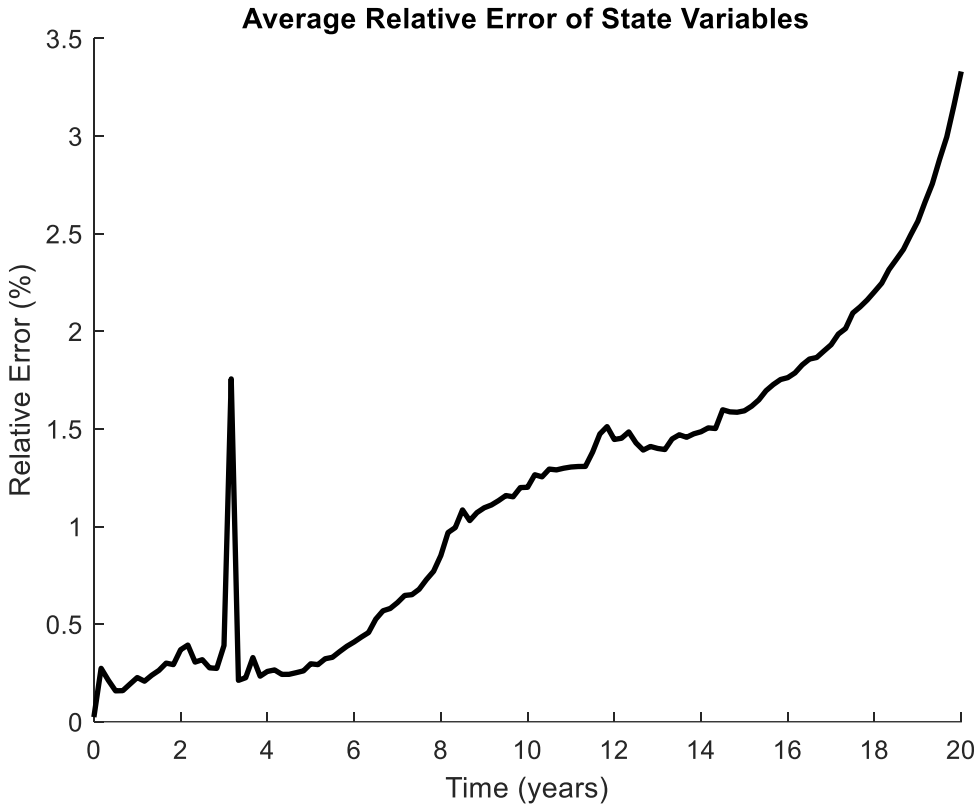


Figure 2.11 Average relative error of all the state variables.

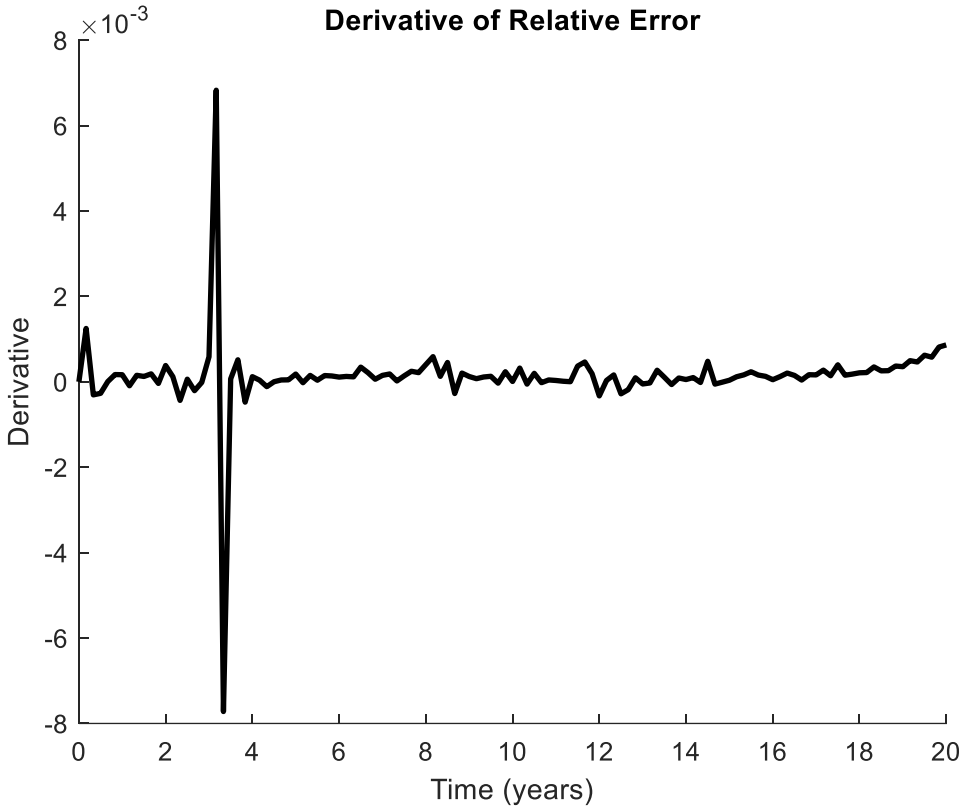


Figure 2.12 Derivative of relative error of all the state variables estimated numerically.

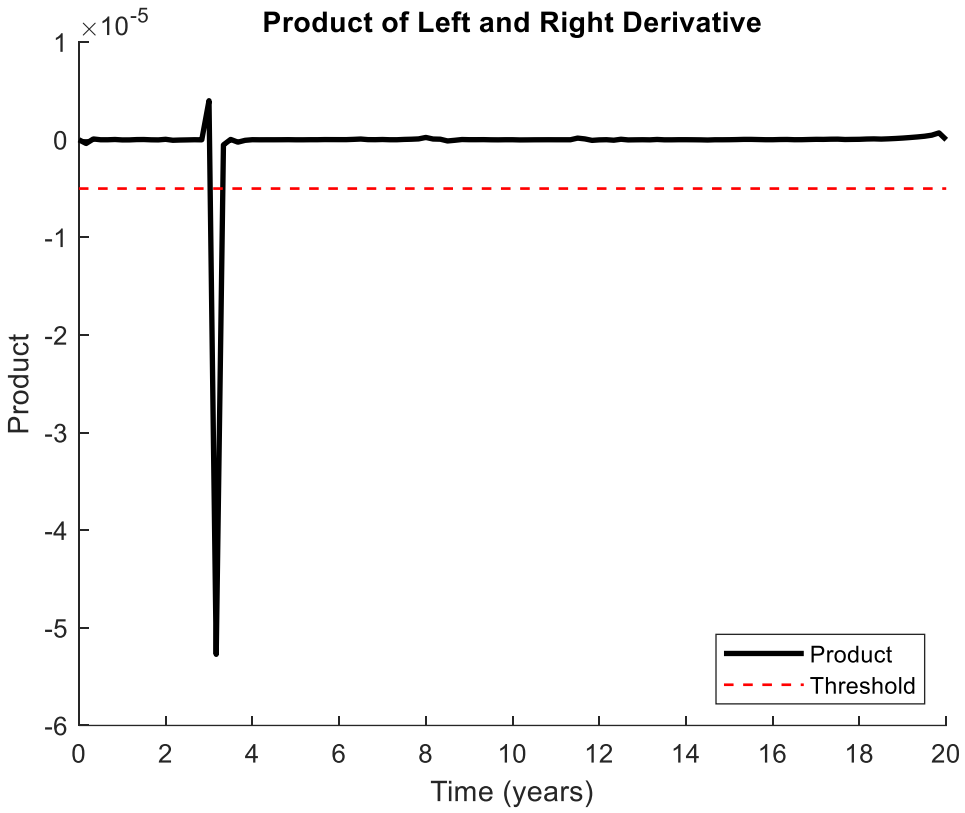


Figure 2.13 Product of the left and right derivative of the relative error at every time point.

An automatic method based on the numerical derivative of the relative error on the validation set is used to identify these turning points. The slope between each two continuous points is computed as the estimation of the derivative of relative error. Then, the product of the left and right derivative at each time point is computed which is shown in **Figure 2.13**. When the average relative error reaches to its local highest point (the spike in **Figure 2.11**), the product of the left and right derivative becomes a negative value, because the left derivative is positive, and the right derivative is negative. Then, the time point which has the most negative value is the most possible point where the trend of the variation of state changes. Here, a threshold value must be set up to filter those points which are slightly negative due to normal vacillation of relative error.

Knowing the possible turning points of state, we can then retrain a group of surrogate models after every turning point to better capture the non-linearity of the variation of state. For each iteration, we only need the snapshots after the turning point and repeat step 2 and step 3 to obtain a new surrogate model. This procedure is performed for as many times as the number of turning points identified above. Now, we have a group of surrogate models, and then they can be used to predict the states for other set of controls (input). The number of surrogate models obtained so far is equal to the number of turning points plus one. The first surrogate model obtained from training with the whole snapshots-matrix is used to capture the general variation trend of the state, and the following ones aim at revising the model whenever the trend of the variation of the state changes. The method of using these surrogate models as the proxy of the high-fidelity reservoir simulation model for the purpose of prediction is shown in detail in step 5.

2.5. Step 5: Test the performance of the surrogate model

The initial condition of the test set $s_1^{(c_t)}$ is the input of the proxy trained from step 1 to step 4, where c_t denotes the control of the testing set. The algorithm of computing the snapshots-matrix S with the proxy and the input $s_1^{(c_t)}$ is shown in **Figure 2.14**. Then, S is compared with the snapshots-matrix collected from the high-fidelity model $X^{(test)}$ to evaluate the performance of the whole NIROM method. The results of estimation are shown in detail in the chapter of Application of the methodology.

Testing:

Data: $s_1^{(c_t)}, R, W(i), \Delta(i), D_p(i), \phi(i)$, where $i \in [0, length(t_{turn})]$

Result: S

$R = R(:,1:end-1,:)$

$S(:,1) = s_1^{(c_t)}$

$r_1 = \phi^T * s_1^{(c_t)}$

$rCurrent = r_1$

$redS(:,1) = rCurrent$

for $k = 0: length(t_{turn}) - 1$

  for $t = t_{turn}(k): t_{turn}(k+1) - 1$ /* $\boldsymbol{t_{turn}(0) = 1}$ */

    for i = 1: o

      for j = 1: $(n_t - 1) * u$

        d = compute distance between $x(i,:)$ and $x(j,:)$ with eq. 5

        $f(i,1) = f(i,1) + [W(k)](i,j) * \beta(d)$

      end

    end

    $rNext = rCurrent + f$

    $redS(:,t+1) = rNext$

    $rCurrent = rNext$

  end

  $S\big(:, t_{turn}(k)+1: t_{turn}(k+1)\big)$ ...

  $= [\phi(k)^t\ ]^{-1} * redS(:, t_{turn}(k)+1: t_{turn}(k+1))$

end

return S

Figure 2.14 algorithm of computing the snapshots-matrix S with the proxy and the input $s_1^{(c_t)}$

3. APPLICATION OF THE METHODOLOGY

3.1. UNISIM-I-D model

In the first case study, the NIROM method is applied on the UNISIM-I-D model to obtain the pressure and oil saturation field. To set up the model, the four production wells are producing at constant flow rate at first and then switch to produce at constant bottom-hole pressure when the reservoir pressure is not high enough to support constant oil flowrate production. Three different selections of timestep and timestep interval used to evaluate the performance of the NIROM method are 60 timesteps * 4 months = 20 years, 120 timesteps * 2 months = 20 years, 240 timesteps * 1 month = 20 years. The performance of the proxies trained by different timestep selections are compared.

One thing to be clarified is that all the four wells are drilled on the left side of the fault and the spread of pressure front and saturation front is stopped by the fault. Thus, the grids on the right of the fault will not be affected and the pressure and saturation on the right side keep unchanged. The NIROM method can capture this phenomenon perfectly. However, in this thesis, the states of the grids on the right side of the fault are not shown in the pressure map and the saturation map, and it is not included in the error calculation in order to better evaluate the ability of NIROM to model changing properties.

Firstly, the results without validation (without implementing the step 4 in the chapter of Methodology) are shown, and then, the results of using validation set to improve the performance of the proxy in predicting multiple variation trends of the states is shown. The two sets of results will be compared and discussed afterwards.

3.1.1. NIROM method on UNISIM-I-D model without validation

In **Table 3.1**, the 5 different sets of controls of oil production rates used for generating the training set are listed under the column of training set initial oil production rate, and the oil production rates used for testing the performance of the NIROM method is listed under the column of testing set initial oil production rate. The training set is used as input to train the proxy of the high-fidelity reservoir simulation model following the procedure introduced in the chapter of Methodology. Then, the proxy is used to predict the snapshots-matrix with the input of the initial condition of the states of the testing set, and the snapshots-matrix is compared with it generated by the high-fidelity model to evaluate the accuracy of the NIROM method.

In **Table 3.2**, the POD order (the number of singular values kept) used for each state variable is listed. When POD is implemented, the row dimension of the snapshots-matrix of one state variable is reduced to its corresponding POD order. The reason why the POD order used for the oil saturation is doubled from it used for the pressure is that the variation of oil saturation is usually more complex than the variation of pressure, and it is easier for oil saturation to be influenced by the high heterogeneity of the reservoir. The state variables usually have the most severe variation in the wells, and this needs to be modelled more carefully. Therefore, in order to make sure that the information of the state variables from the well is kept while training, the snapshots of the state variables in well grids are taken separately from the non-well grids.

Table 3.1 The initial oil flowrates for the training set and the testing set for each well in the case study of UNISIM-I-D without validation.

Well name	Training set initial oil production rate (m3/d)					Testing set initial oil production rate (m3/d)
NA1A	900	950	1000	1050	1100	925
NA2	900	950	1000	1050	1100	975
NA3D	900	950	1000	1050	1100	1025
RJS19	900	950	1000	1050	1100	1075

Table 3.2 The POD order for each state variable used for UNISIM-I-D model.

	Non-well grids pressure	Non-well grids oil saturation	Non-well grids water saturation	Well grids pressure	Well grids oil saturation	Well grids water saturation	Well pressure
POD order	10	20	20	10	20	20	10

3.1.1.1. Pressure matching results on UNISIM-I-D without validation

In the results of pressure prediction, the pressure obtained from the NIROM method matches quite well in general with it collected from the high-fidelity reservoir simulation model. **Figure 3.1** shows the pressure matching results for the three different timestep setups in normal grids (not passed through by a well) and well grids. The pressure predicted by the NIROM method has captured the general trend in both normal grids and well grids. There are two obvious turning points in the pressure curve. One is at around 2 years because of the transfer from transient flow to pseudo-steady state flow, and the other is at around 12 years due to the switch of production strategy from constant oil flowrate production to constant bottom hole pressure production. The pressure curve predicted by the NIROM method is a little off from the pressure curve obtained

from the high-fidelity model at these turning points. This issue is improved when shorter timestep interval is used.

To quantify the accuracy of the NIROM method, relative error is computed for each state variable. **Figure 3.2** shows the average relative error of the pressure of normal grids. As we can see, the relative error reaches to the local highest point at around 2 years when the flow pattern switches.

Generally, the NIROM method has better performance in pressure prediction when using shorter timestep interval for generating the training set. This is because shorter timestep interval and more timesteps taken in the training set means more detailed information about the model. This behavior is shown clearly in **Figure 3.2**. The case using the shortest timestep interval has the highest accuracy (within 3.22% average relative error), and the case using longest timestep has the worst accuracy (within 11.06% average relative error).

Figure 3.3 shows the average relative error of the pressure for well grids. Compared with **Figure 3.2**, the pressure in the well grids predicted by the NIROM method has higher relative error. The average relative error reaches to 10% and to 23.26% in the best case (240 steps * 1 month) and the worst case (60 steps * 4 months). Even though POD is implemented to the snapshots-matrix of the well grids and the normal grids separately, and the energy percentage of POD is high enough to guaranty that most information of the principle components related to the wells is reserved in the reduced snapshots matrix obtained from POD, the performance of the NIROM method in predicting the pressure in the well grids is still worse than it in normal grids. This is because that the well grids have more rapid and obvious reaction regarding the switch of production strategy, and there is a sharper turning point in the pressure curve in the well grids. It is hard for the proxy to capture such detail in the well.

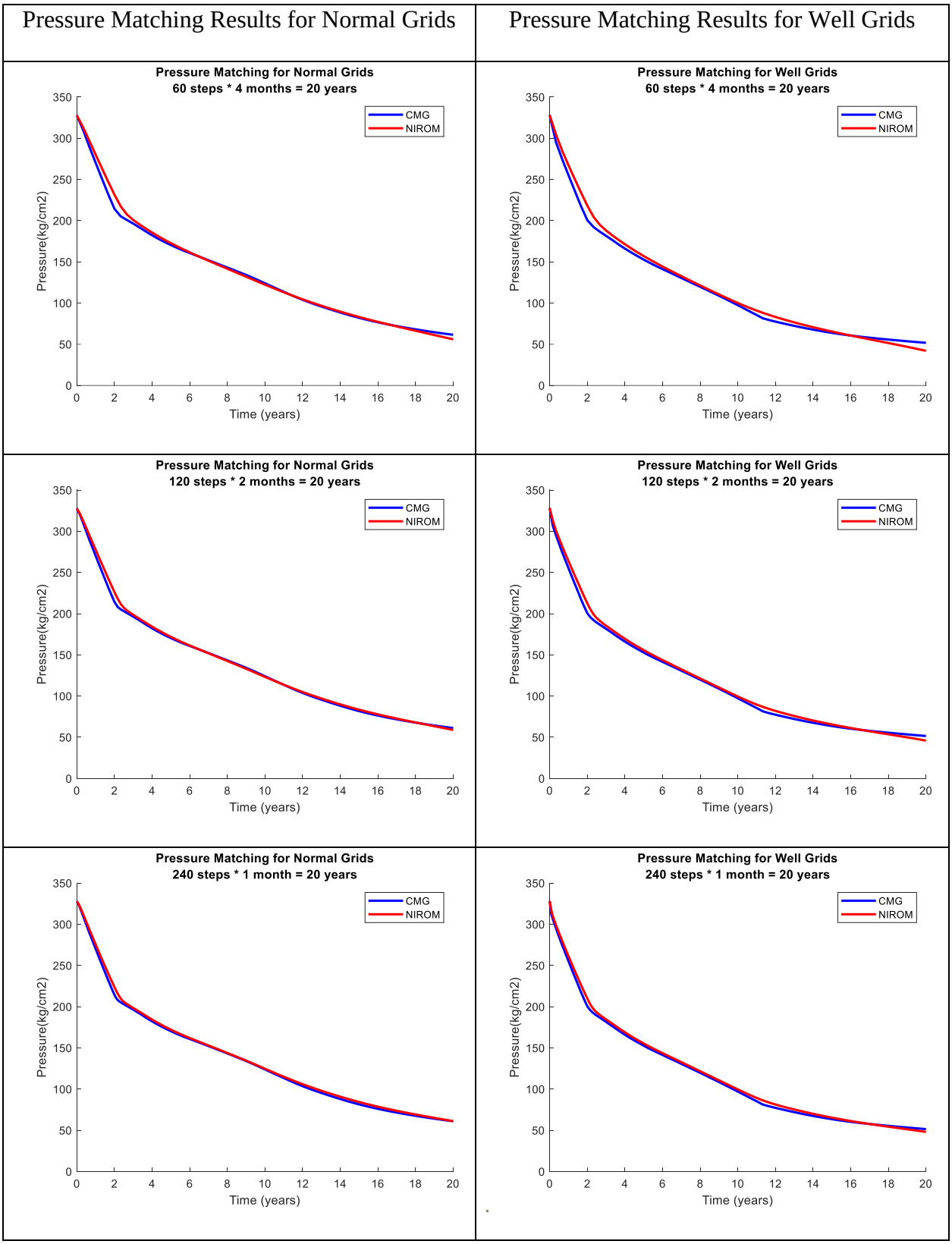


Figure 3.1 Pressure matching results of 3 different selections of timestep interval for UNISIM-I-D without validation.

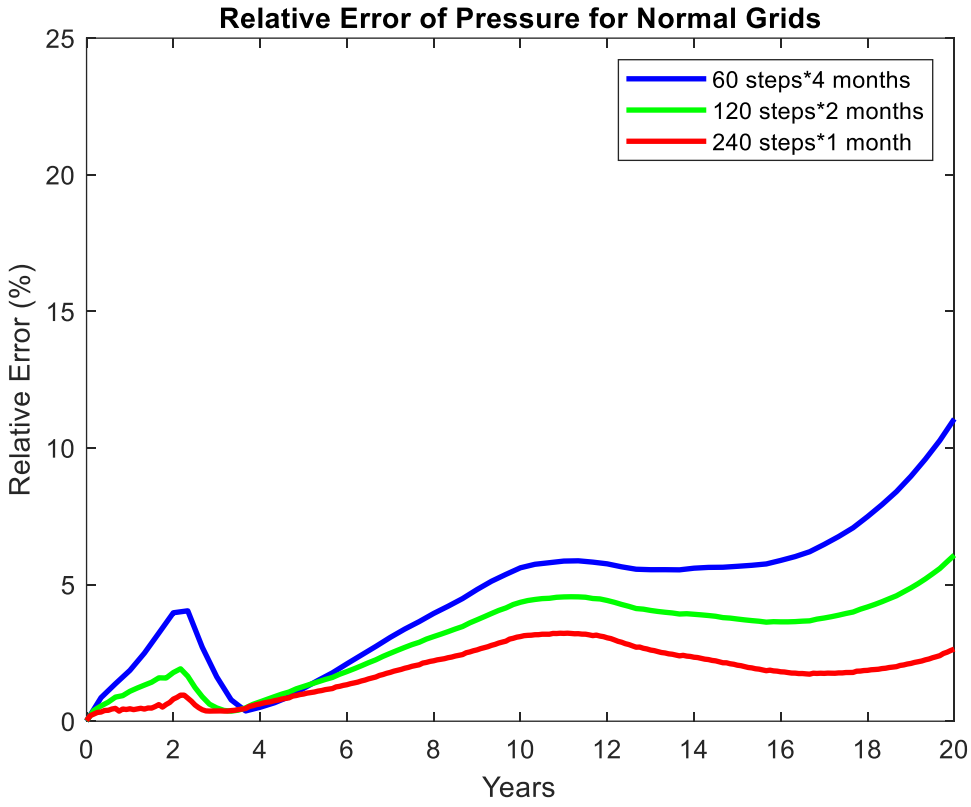


Figure 3.2 Average relative error of pressure for normal grids in UNISIM-I-D without validation.

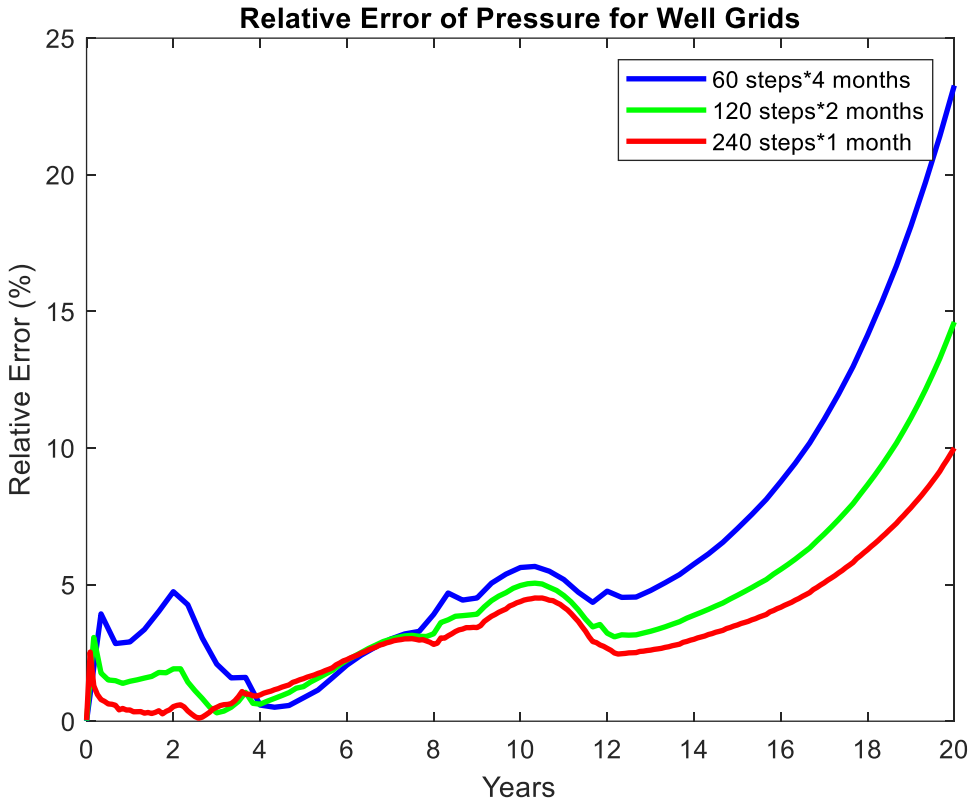


Figure 3.3 Average relative error of pressure for well grids in UNISIM-I-D without validation.

Figure 3.4 shows the pressure map of the first layer of UNISIM-I-D at the end of the simulation (20 years) obtained from the high-fidelity reservoir simulation model (CMG) for the test set. The pressure maps predicted by the NIROM method with the three different timestep selections are shown in the first column of **Figure 3.5**. The relative error of each grid is shown in the second column in **Figure 3.5**. In the pressure maps in the first column, the NIROM method has captured the location of the pressure front in all three timestep selections. In the relative error maps in the second column, the highest relative error concentrates around the four production wells. This is because of the phenomenon mentioned above that the well grids have a more rapid and obvious reaction to the switch in production strategy, and it is hard to be captured by the NIROM method. This problem is partly solved by using shorter timestep interval and more timesteps while training,

which has improved the overall accuracy of the proxy obtained. Although the relative error near the wells is still higher than it in the normal grids, it is within an acceptable range (within 10 % shown in **Figure 3.3** when using the shortest timestep interval for training).

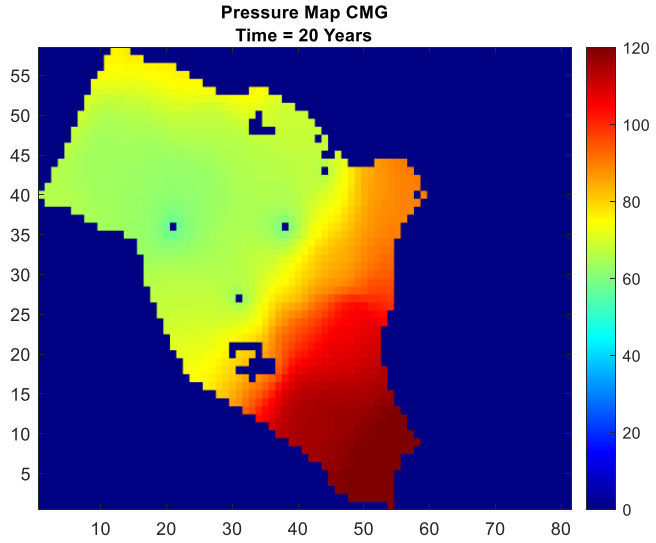


Figure 3.4 Pressure map of layer 1 of UNISIM-I-D at 20 years obtained by high-fidelity reservoir simulator.

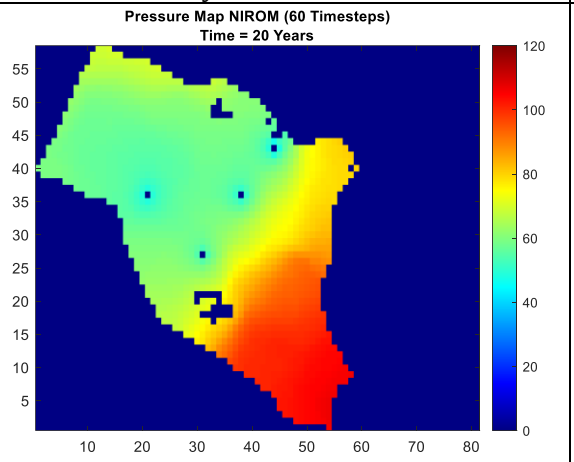
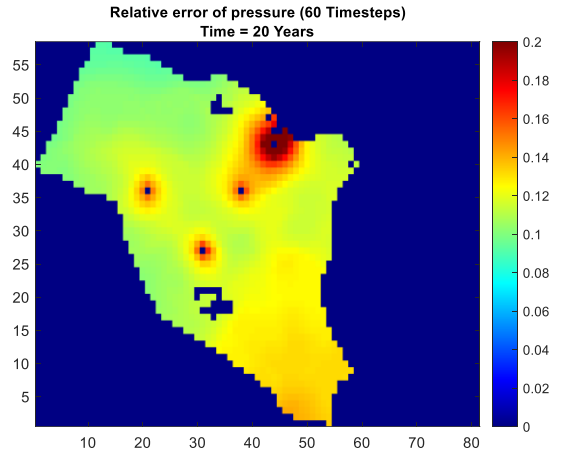
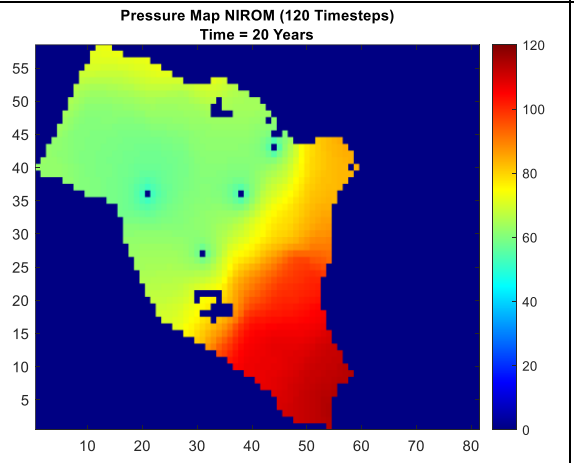
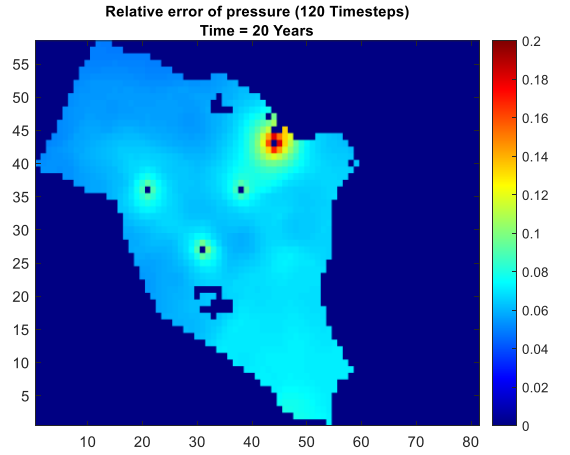
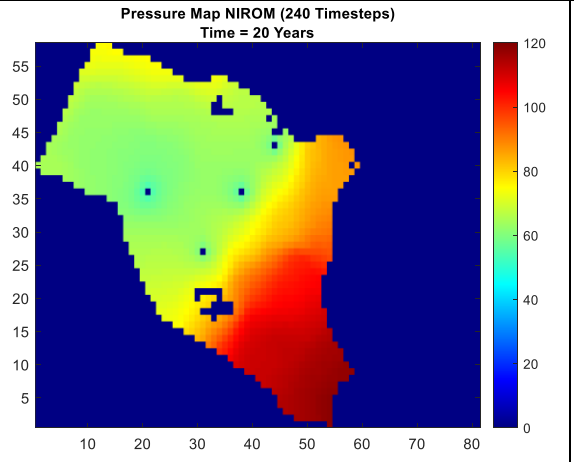
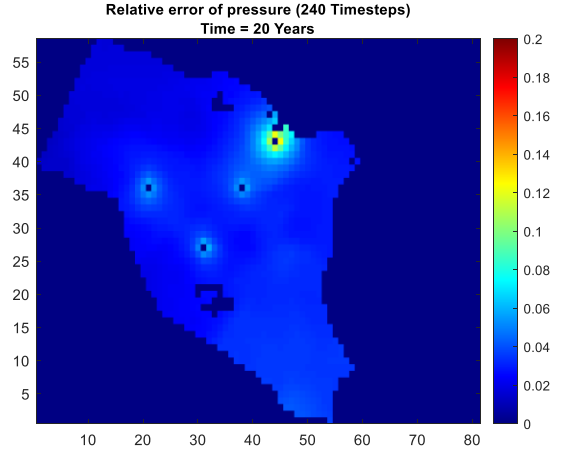
Pressure Map of the First Layer of UNISIM-I-D Obtained by the NIROM Method	Relative Error of Pressure in the First Layer of UNISIM-I-D
	
	
	

Figure 3.5 Pressure map of layer 1 of UNISIM-I-D at 20 years obtained by the NIROM method without validation for 3 different selections of timestep intervals(left column). The relative error of pressure for each grid (right column).

3.1.1.2. Oil saturation matching results on UNISIM-I-D without validation

Figure 3.6 presents the matching results of oil saturation obtained from the NIROM proxy and from the high-fidelity simulator for the three different timestep selections. The oil saturation prediction obtained by the NIROM method is not as accurate as the pressure prediction. Before the oil saturation front reaches to a grid, the oil saturation in the grid holds constant for some time, the proxy has captured this behavior well. However, right after the oil saturation front reaches to the grid, where the oil saturation starts to decrease sharply, the saturation is overestimated by the NIROM method.

The proxy has relative worse performance when multiple variation trends of the state variable exist. The variation curve predicted by RBF algorithm tends to be smoother compared to the curve obtained by the high-fidelity model. This is because the prediction using RBF algorithm is affected by all the data points in the training set basing on the distance. The data points coming from one of the trends may be far away from the data points from other trends in distance, which means the effect of them has been minimized by the radial basis function automatically. However, these points from a different trend still offset some influence of the data points in the correct trend to the prediction result due to cumulation effect.

The oil saturation matching results for normal grids shown in **Figure 3.6** is a good example of this characteristic of the RBF algorithm. In the training set, the data points can be separated into three groups corresponding to the three different trends in oil saturation (e.g. 1. the phase before the saturation front reaches to the grid, 2. after the saturation front reaches to the grid and the oil saturation has not decreased to the residue oil saturation, 3. after the oil saturation reduces to the residue oil saturation). When computing the snapshots in the second and third trend using RBF algorithm, the data points in the first trend still has influence on the snapshots.

This issue is improved by an algorithm aiming at capturing the multiple variation trends of the state which is introduced in the chapter of Methodology in the section of step 4. The results of implementing this algorithm is presented in the next section.

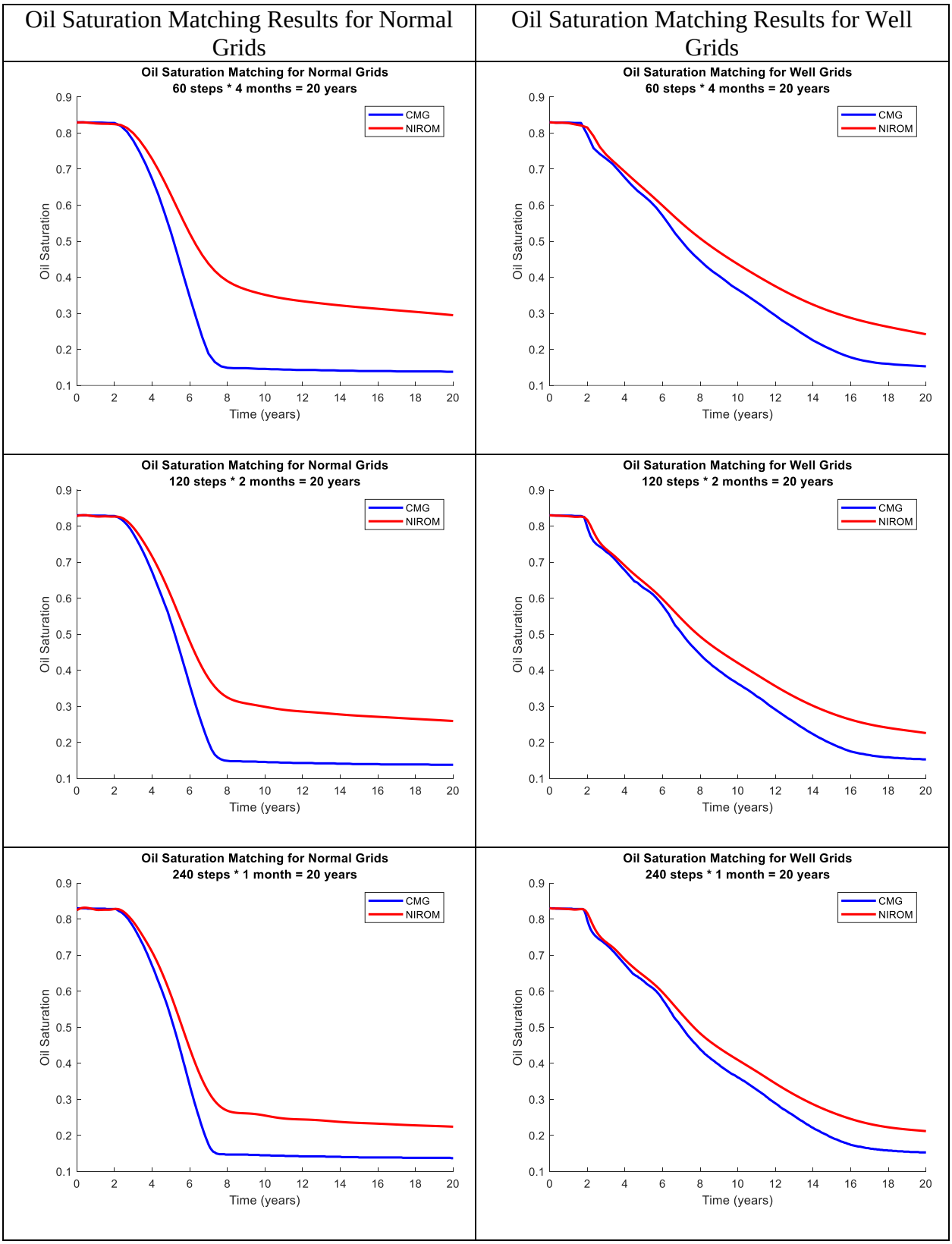


Figure 3.6 Oil saturation matching results of 3 different selections of timestep interval for UNISIM-I-D without validation.

Figure 3.7 and **Figure 3.8** present the average relative error of oil saturation for normal grids and well grids respectively. The case trained by the shortest timestep interval has the highest accuracy in general in both the normal grids and the well grids (within 6% average relative error for normal grids and within 5.45% average relative error for well grids).

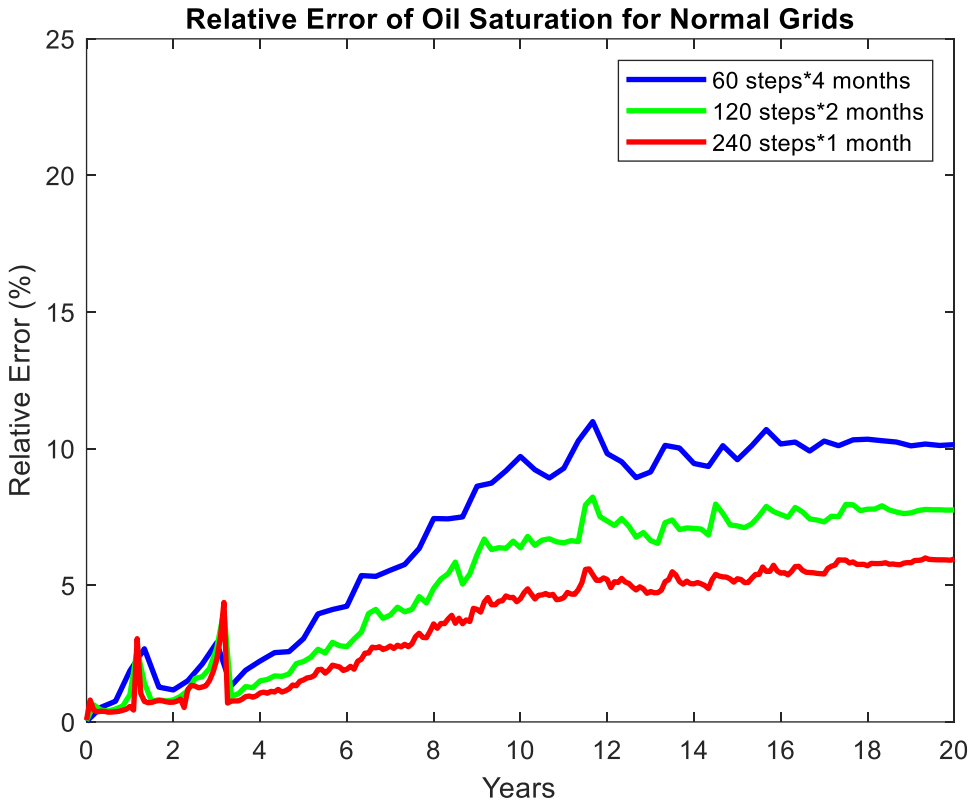


Figure 3.7 Average relative error of oil saturation for normal grids in UNISIM-I-D without validation.

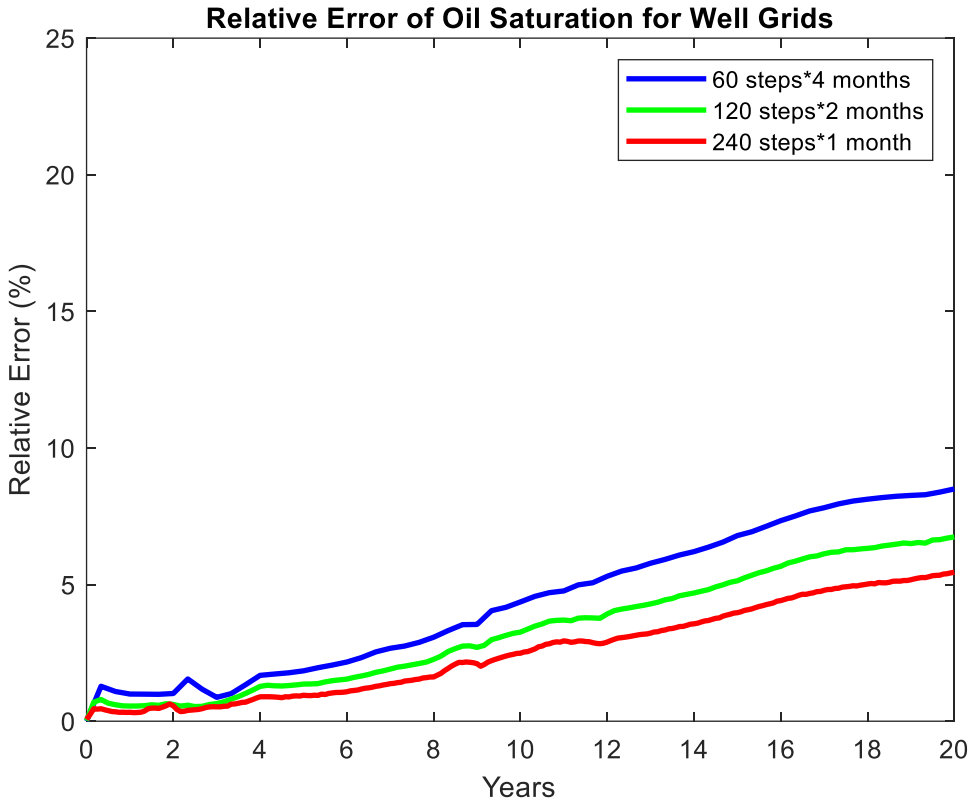


Figure 3.8 Average relative error of oil saturation for well grids in UNISIM-I-D without validation.

Figure 3.9 presents the oil saturation map of the first layer of UNISIM-I-D obtained from the high-fidelity model. In **Figure 3.10**, the first column presents the oil saturation maps predicted by the proxy with the three different timestep selections. The second column shows the corresponding relative error maps. Comparing the relative error maps with **Figure 1.1**, we can see that the highest relative error concentrates around the area which has relative higher permeability. The oil depletes faster in this area which makes the turning between the behaviors of each phase sharper, and thus makes it harder for the proxy to capture.

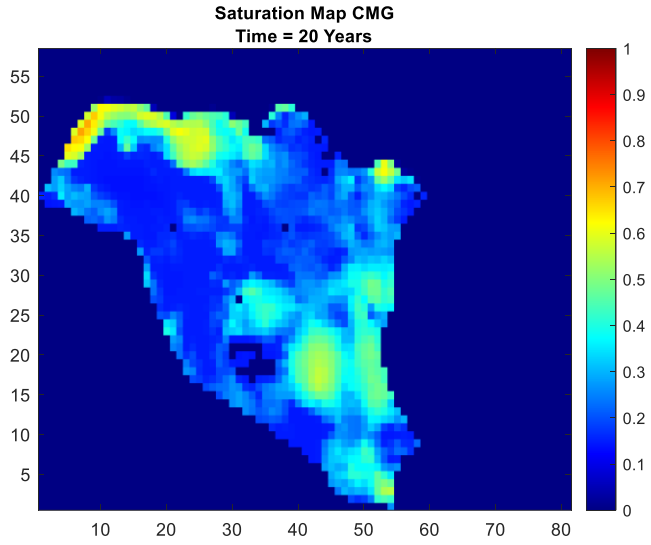


Figure 3.9 Oil saturation map of layer 1 of UNISIM-I-D at 20 years obtained by high-fidelity reservoir simulator.

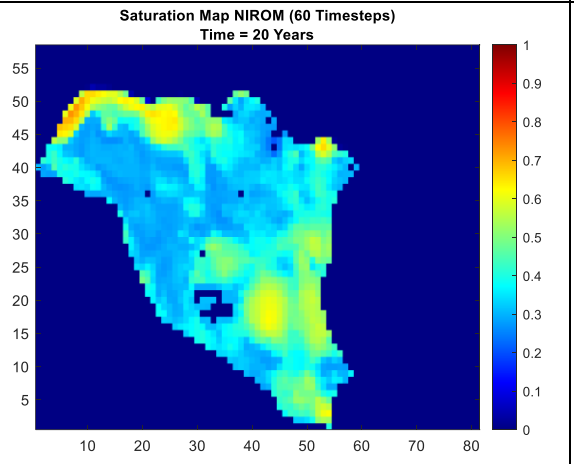
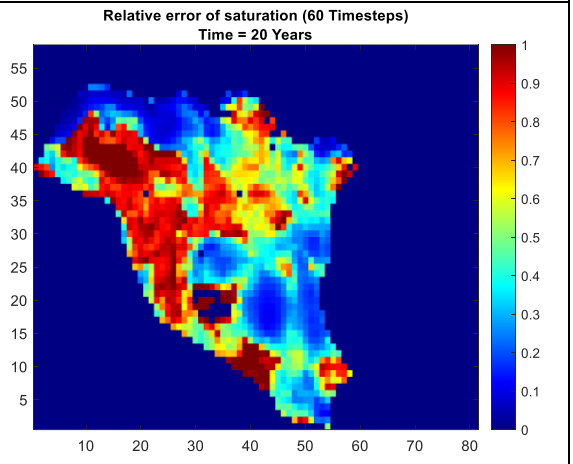
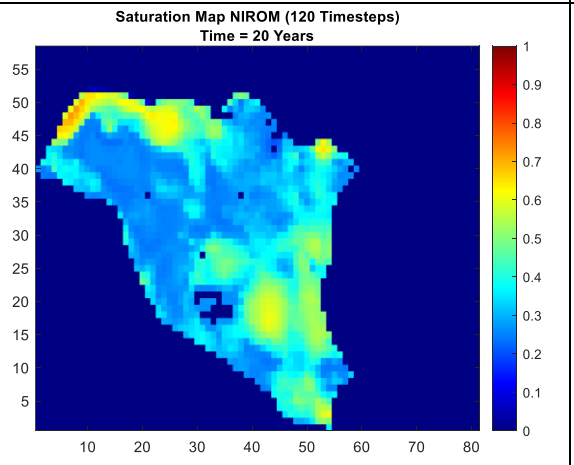
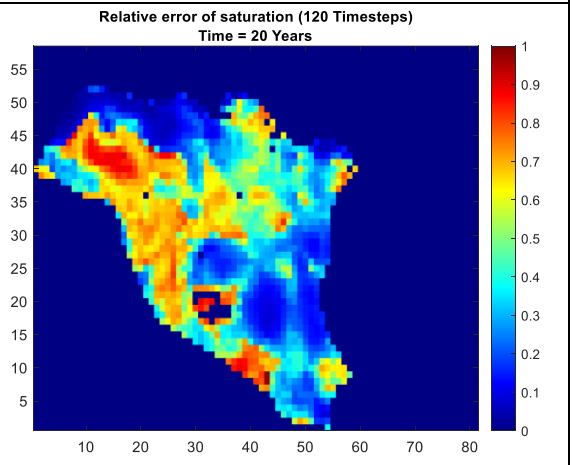
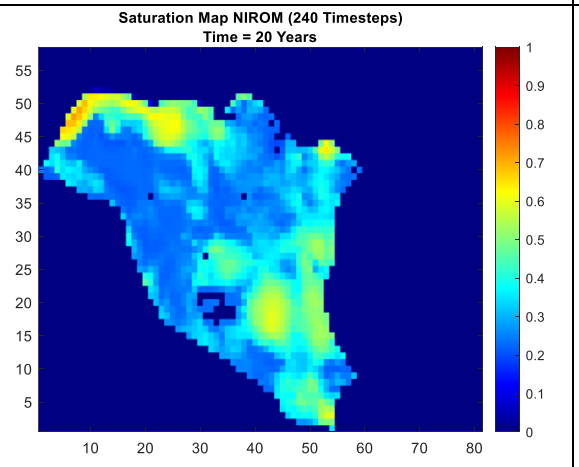
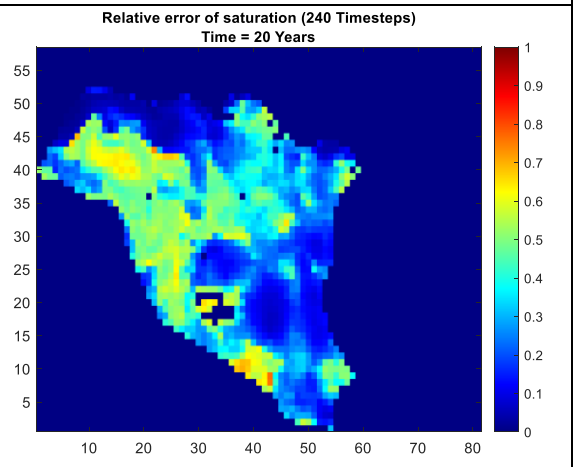
Oil Saturation Map of the First Layer of UNISIM-I-D Obtained by the NIROM Method	Relative Error of Oil Saturation in the First Layer of UNISIM-I-D
	
	
	

Figure 3.10 Oil saturation map of layer 1 of UNISIM-I-D at 20 years obtained by the NIROM method without validation for 3 different selections of timestep intervals(left column). The relative error of oil saturation for each grid (right column).

3.1.1.3. Oil rate matching results without validation

Figure 3.11 to **Figure 3.13** show the oil production rate computed by the NIROM method and the high-fidelity reservoir simulator in the cases of 60 timesteps * 4 months = 20 years, 120 timesteps * 2 months = 20 years and 240 timesteps * 1 month = 20 years respectively. We can see that the NIROM method has generated reasonable oil production rate result for each well in general. However, when the oil production curve reaches to the switching point of constant rate production to constant bottom-hole pressure production, the NIROM method has the worst prediction result. This decrease in accuracy of prediction is improved when using training data set built by shorter timestep interval and will be further improved by using a validation data set.

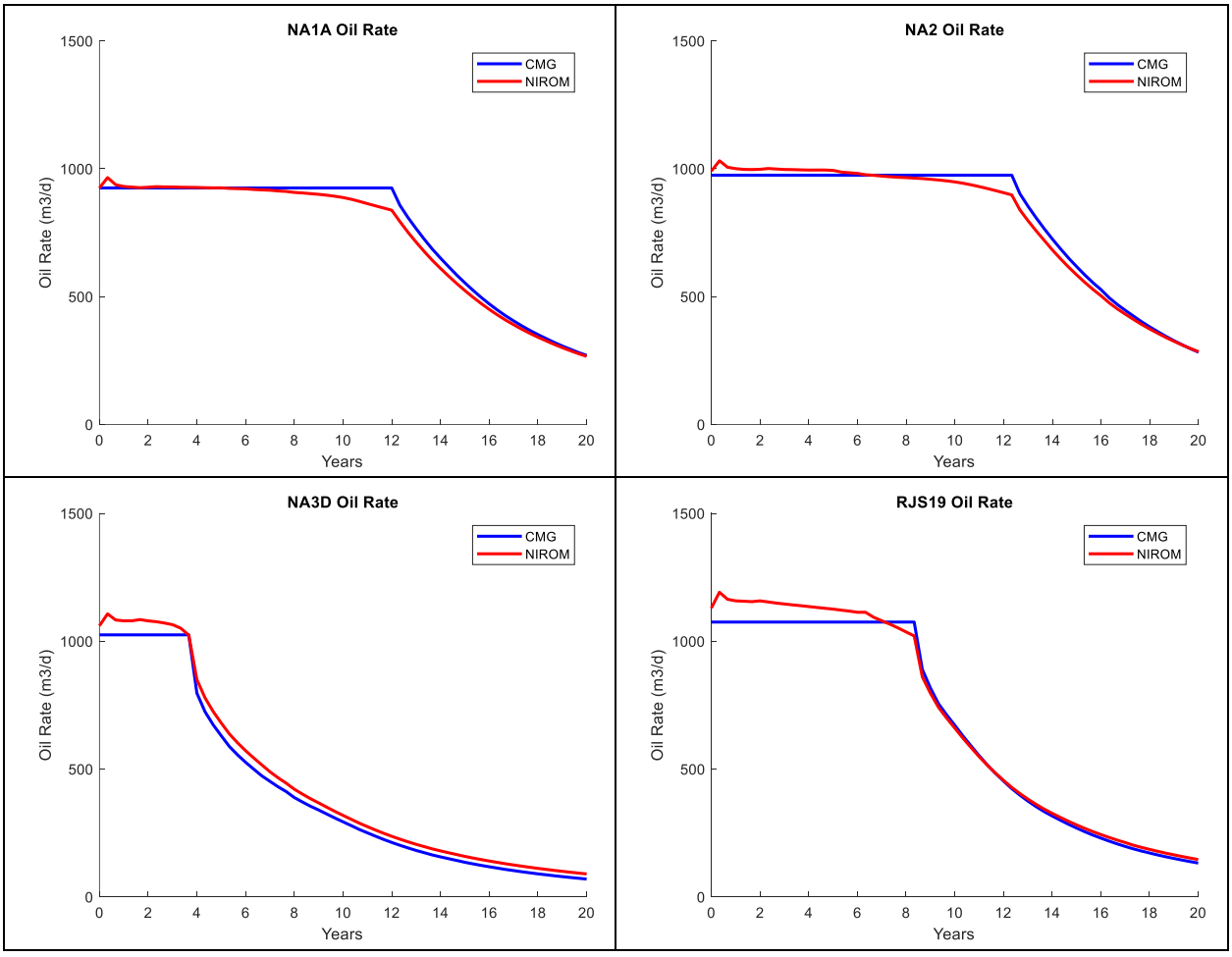


Figure 3.11 60 timesteps * 4 months oil rate matching results for each well (without validation).

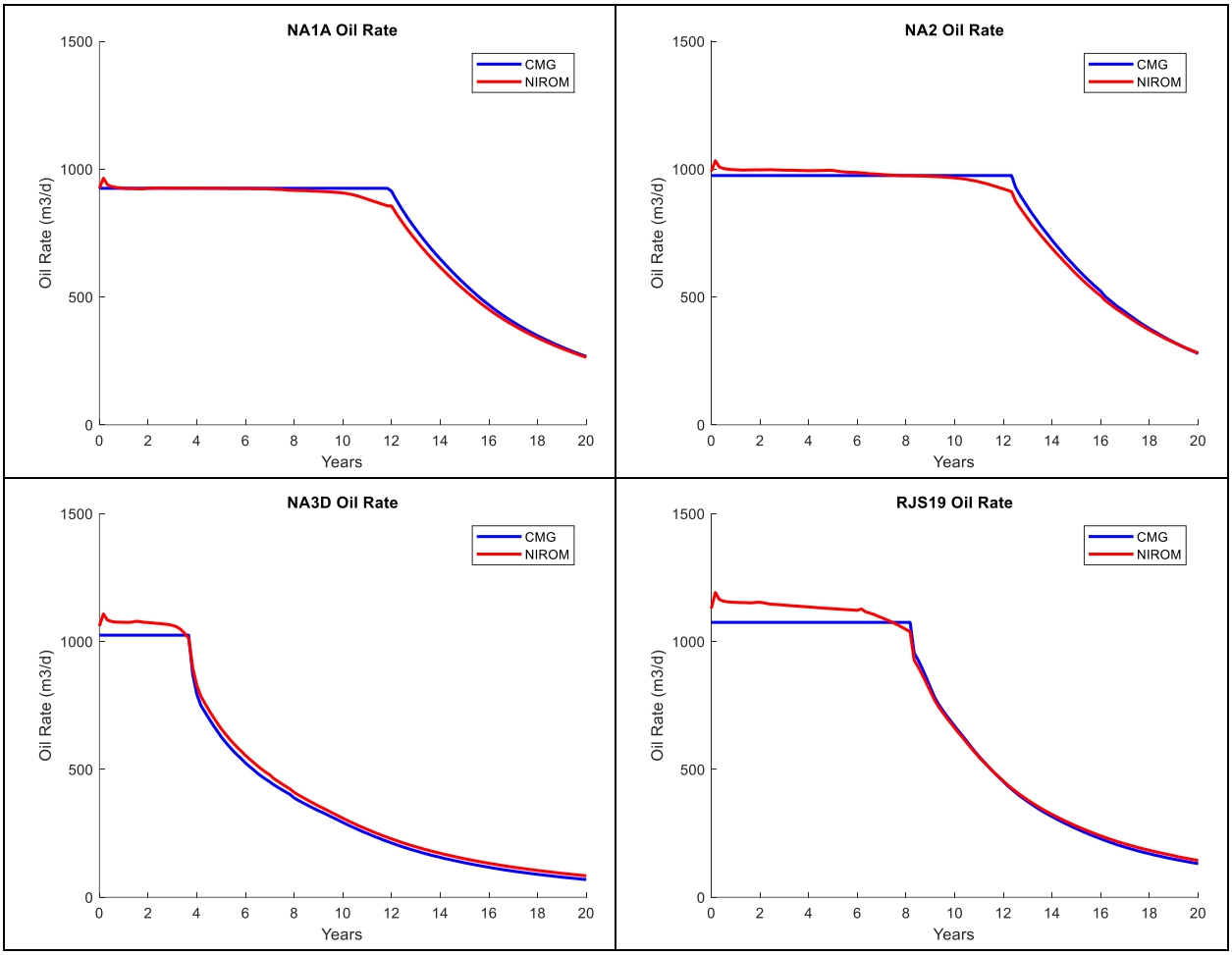


Figure 3.12 120 timesteps * 2 months oil rate matching results for each well (without validation).

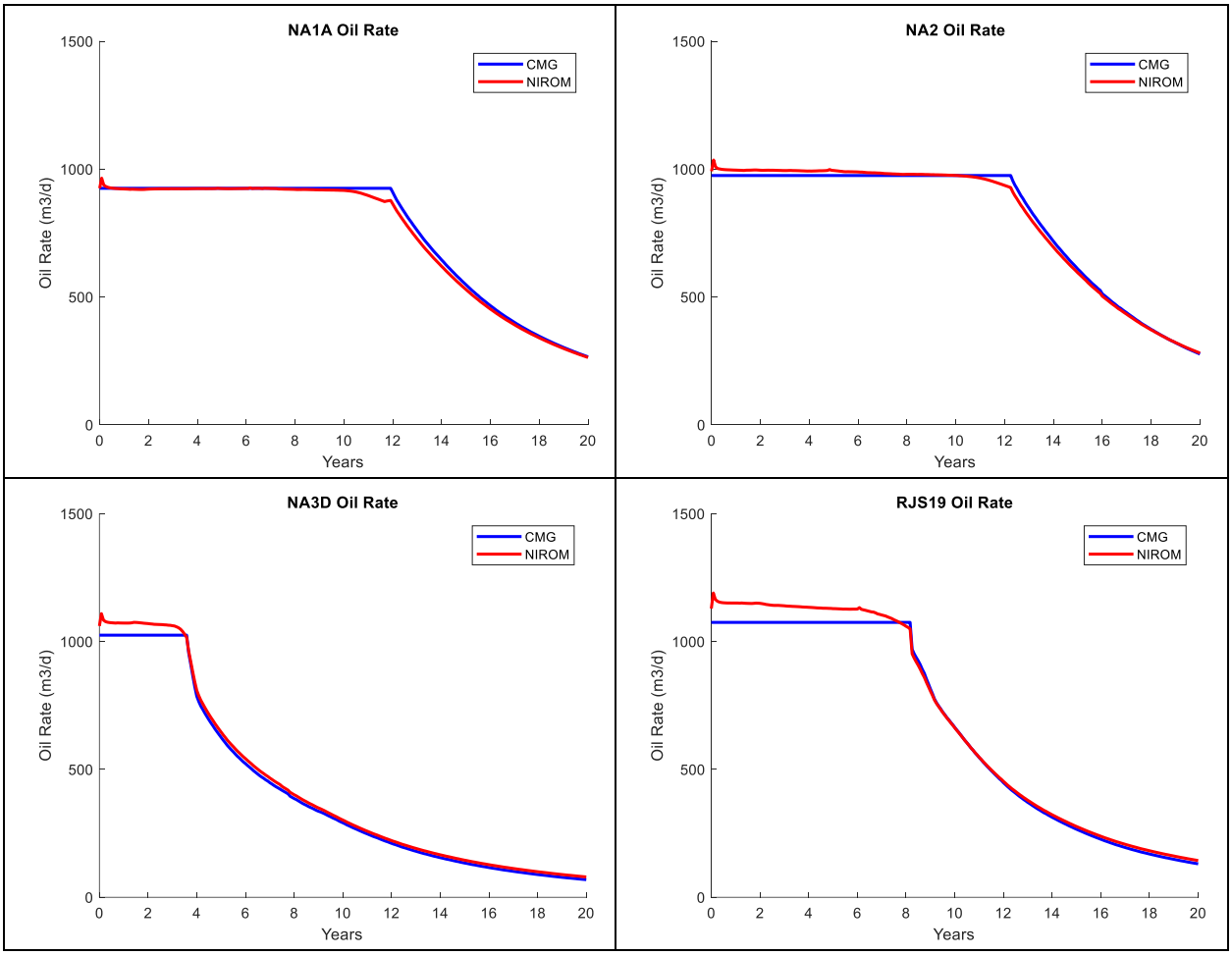


Figure 3.13 240 timesteps * 1 month oil rate matching results for each well (without validation).

Figure 3.14 to **Figure 3.16** show the relative error of oil rate prediction for each well in the cases of 60 timesteps * 4 months = 20 years, 120 timesteps * 2 months = 20 years and 240 timesteps * 1 month = 20 years respectively. We can more clearly see that using training data set built by shorter timestep interval can improve the performance of the NIROM method in oil rate prediction. In the most accurate case, which uses 1 month as the timestep interval, the highest relative error is 5.61%, 6.22%, 15.35%, 10.69% for the well NA1A, NA2, NA3D, RJS19 respectively.

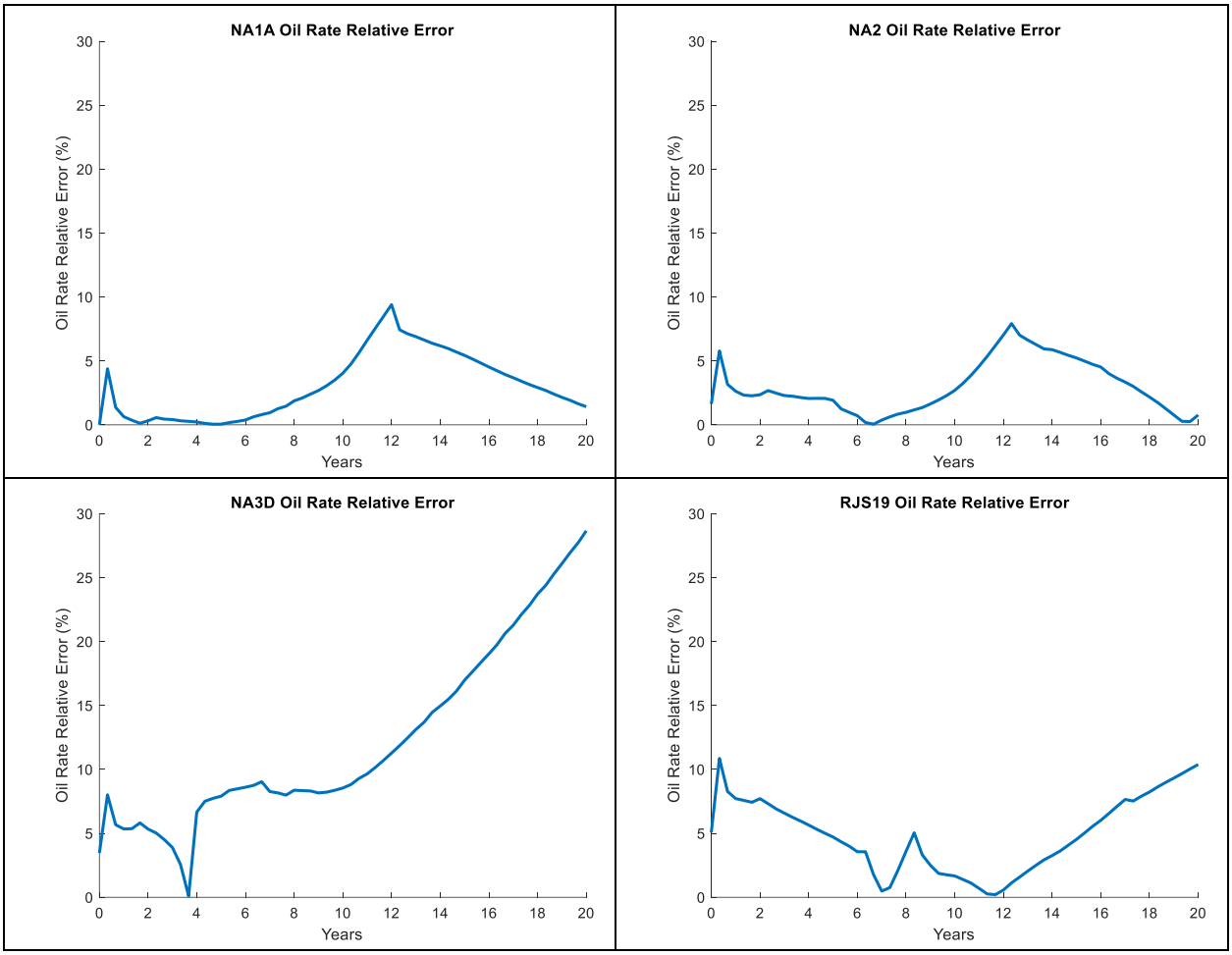


Figure 3.14 60 timesteps * 4 months oil rate relative error (without validation).

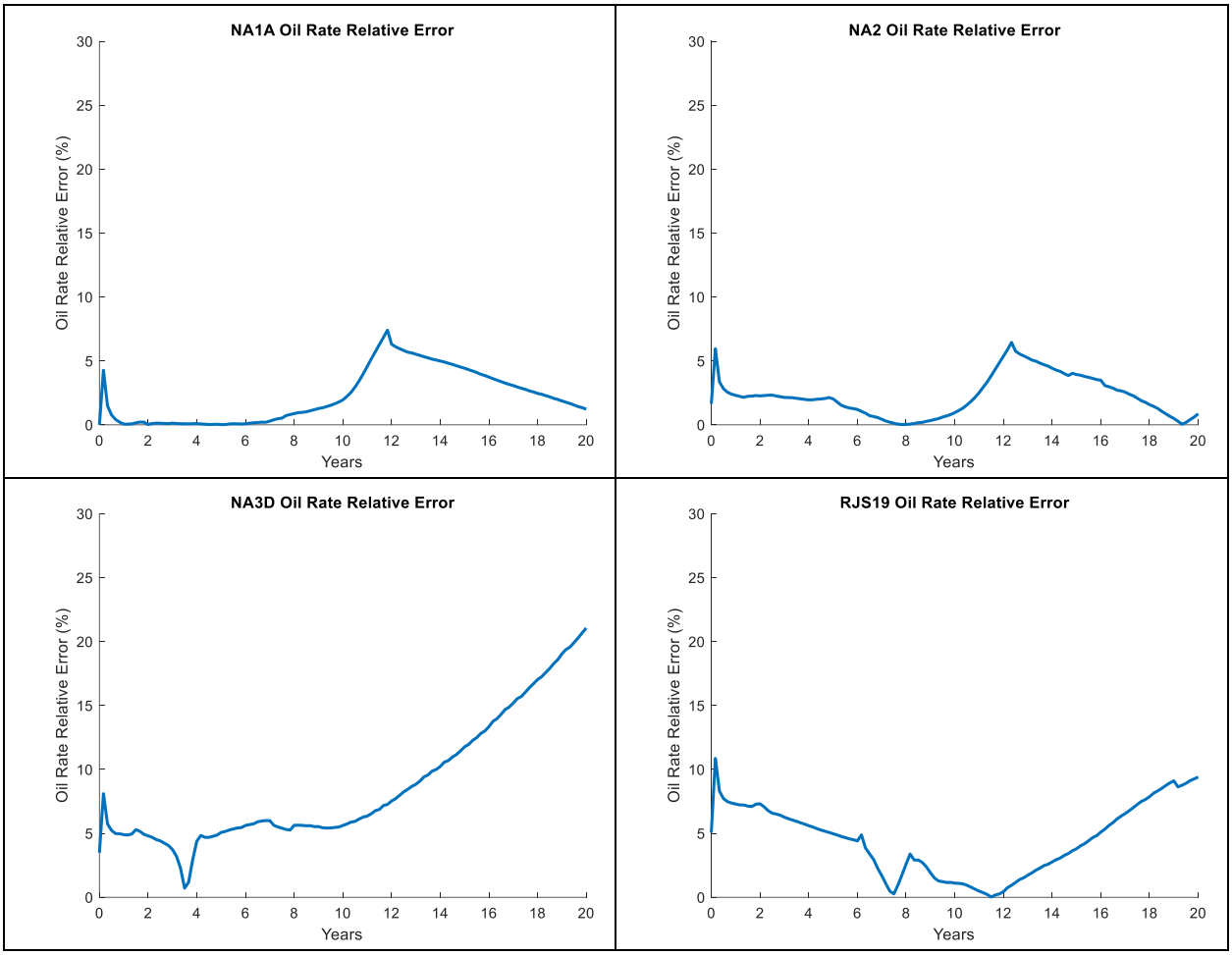


Figure 3.15 120 timesteps * 2 months oil rate relative error (without validation).

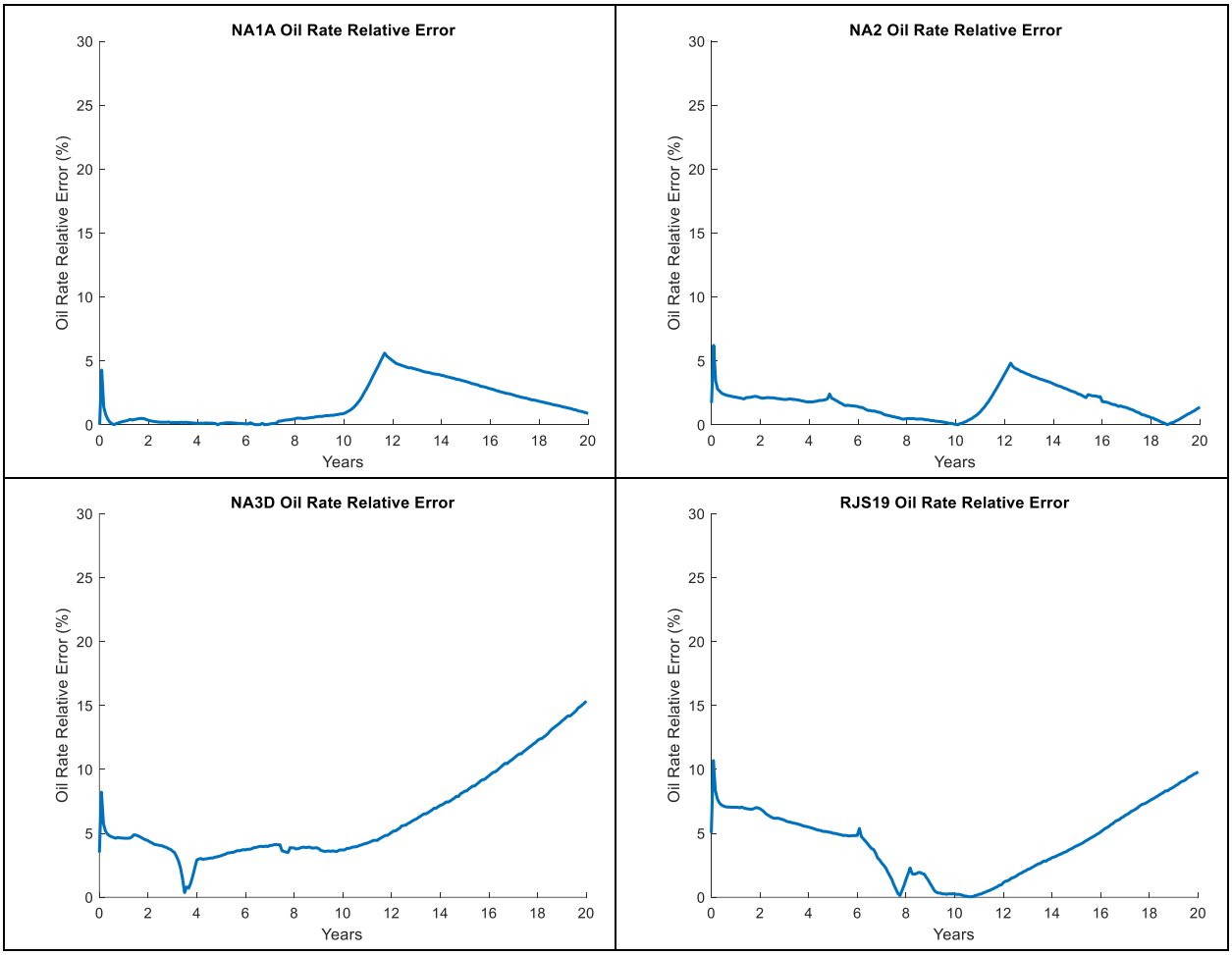


Figure 3.16 240 timesteps * 1 month oil rate relative error (without validation).

Figure 3.17 to **Figure 3.19** show the cumulative oil production volume for each timestep case, and **Figure 3.20** to **Figure 3.22** show the correlated relative error plot. The NIROM method has generated accurate prediction in cumulative oil production and the highest relative error are 2.14%, 3.96%, 5.86%, 8.02% in well NA1A, NA2, NA3D, RJS19 respectively.

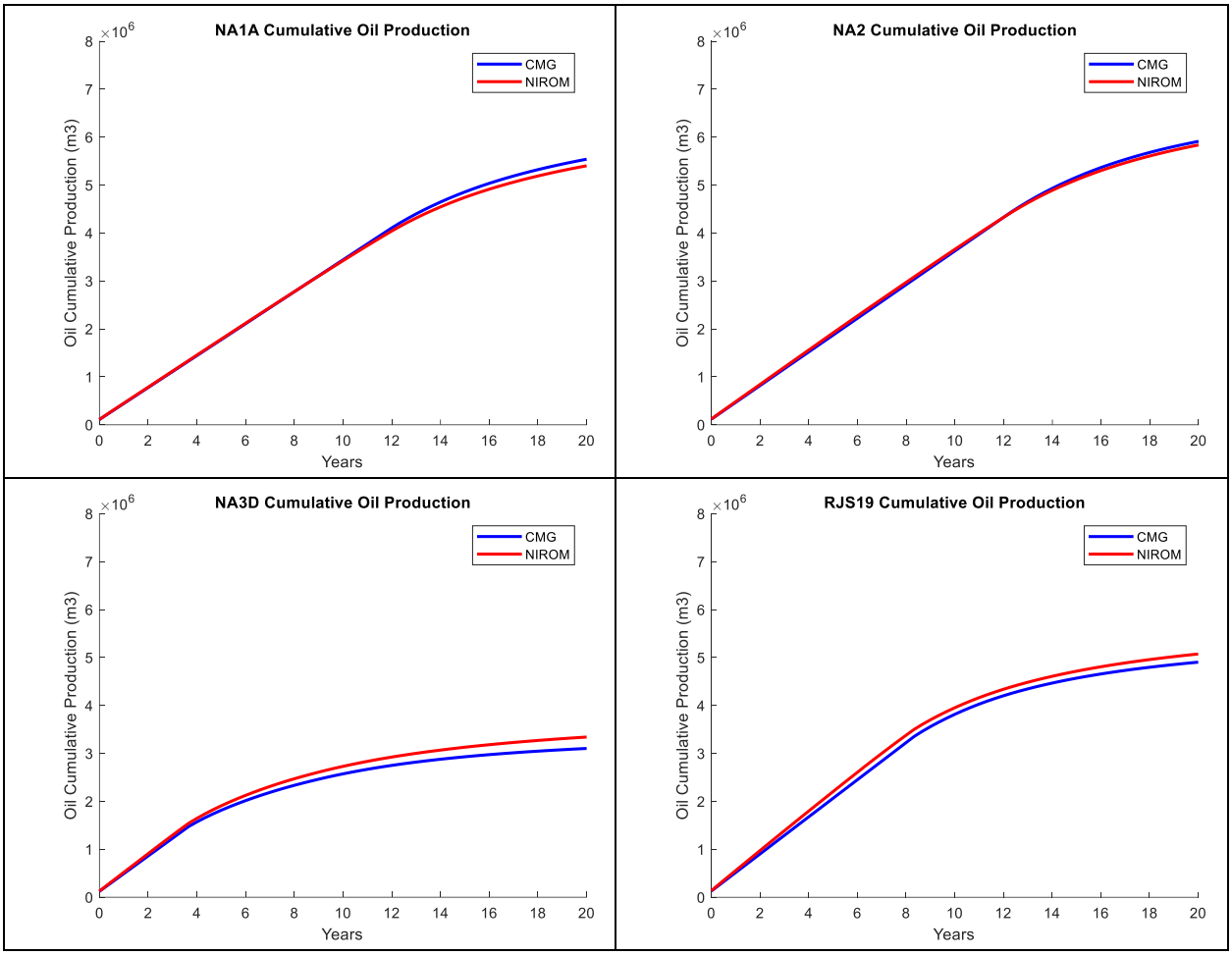


Figure 3.17 60 timesteps * 4 months cumulative oil production volume matching results for each well (without validation).

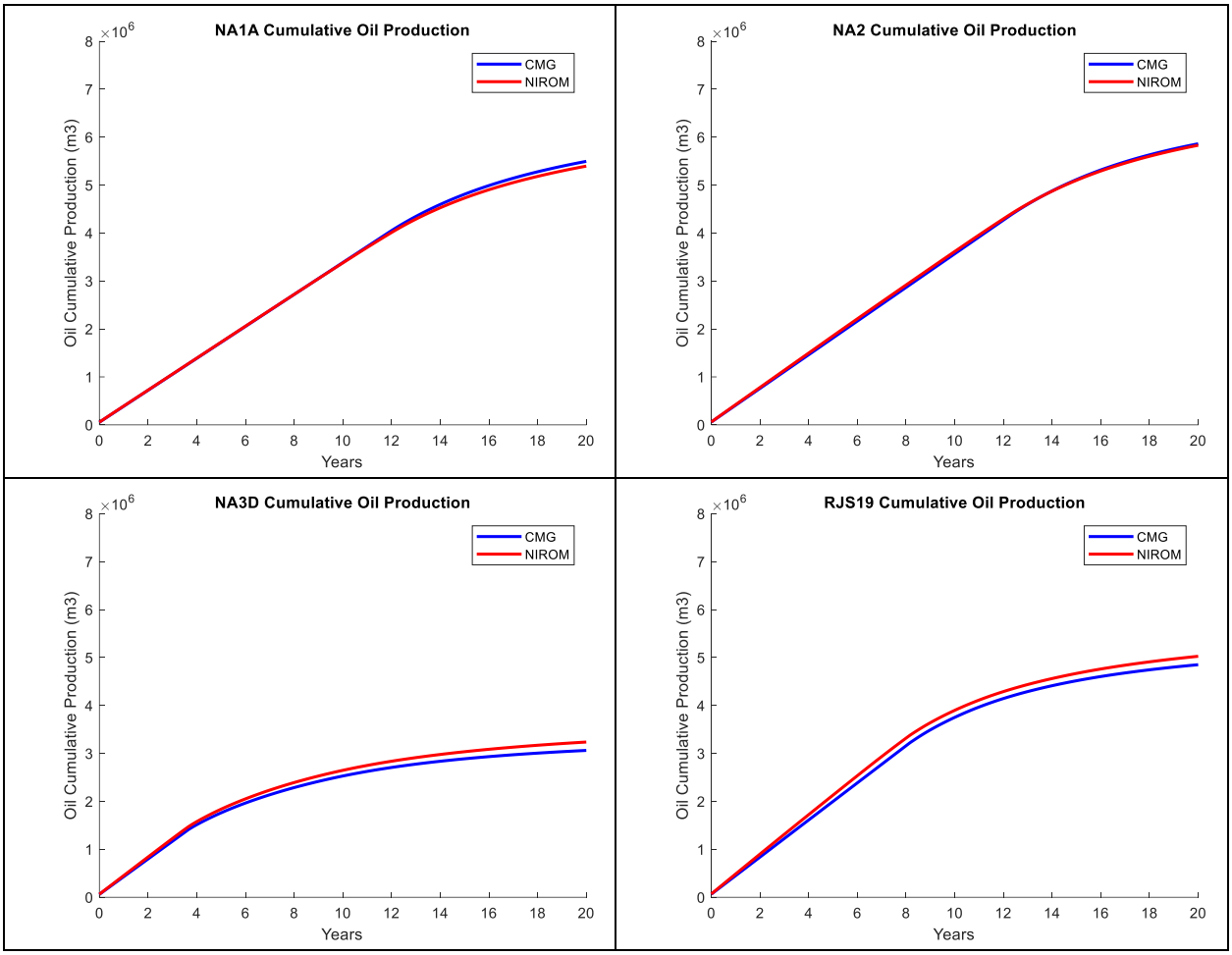


Figure 3.18 120 timesteps * 2 months cumulative oil production volume matching results for each well (without validation).

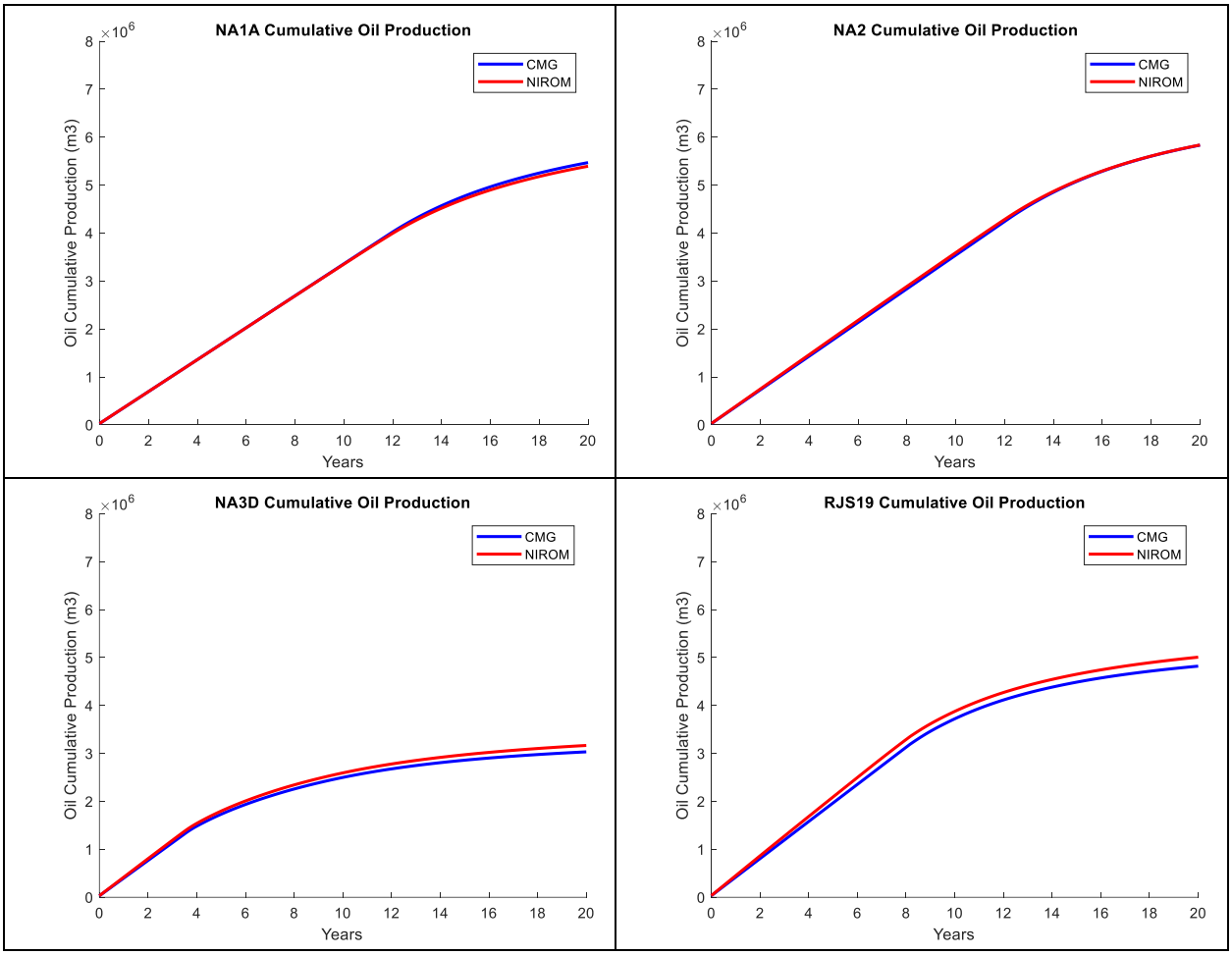


Figure 3.19 240 timesteps * 1 month cumulative oil production volume matching results for each well (without validation).

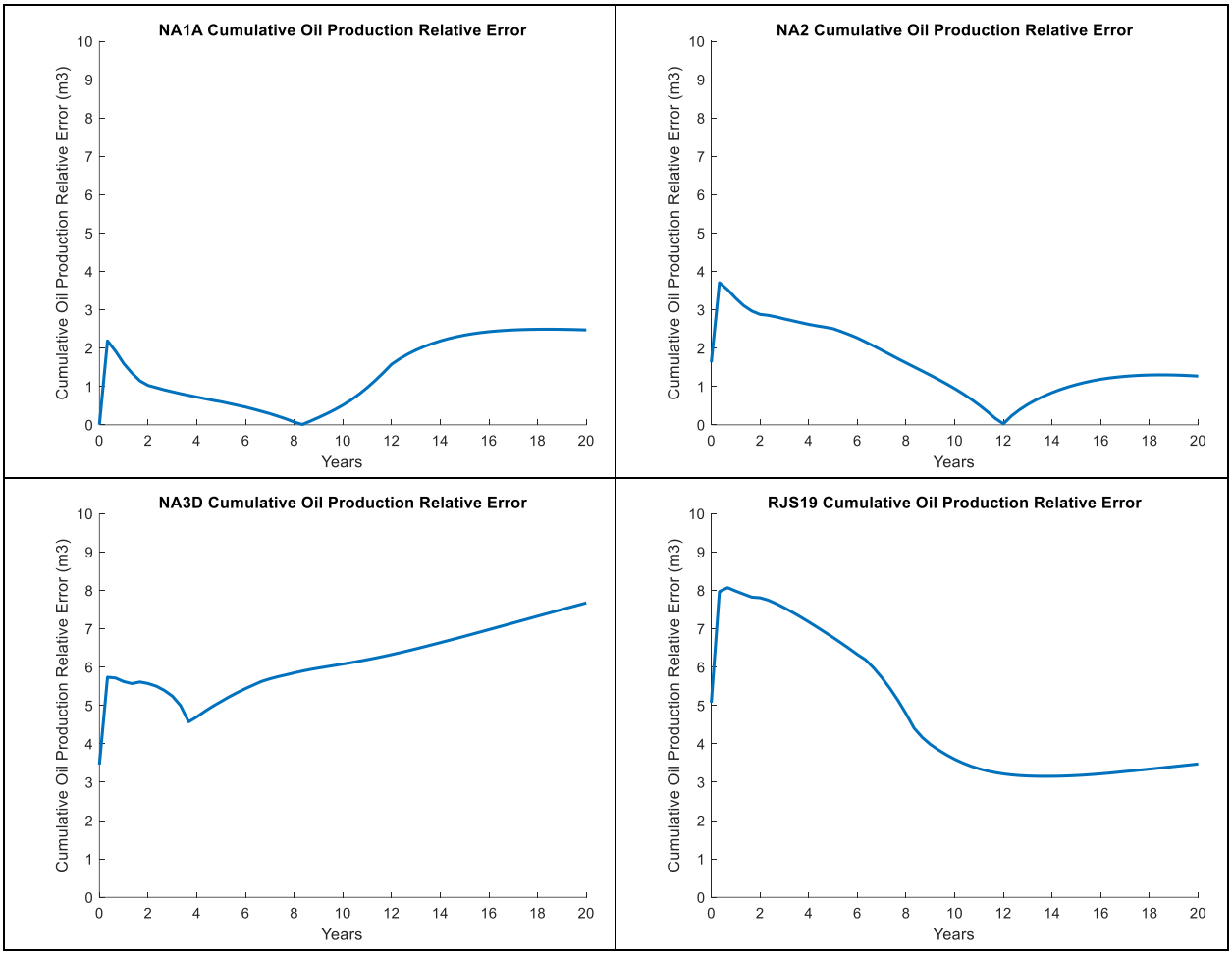


Figure 3.20 60 timesteps * 4 months cumulative oil production volume relative error (without validation).

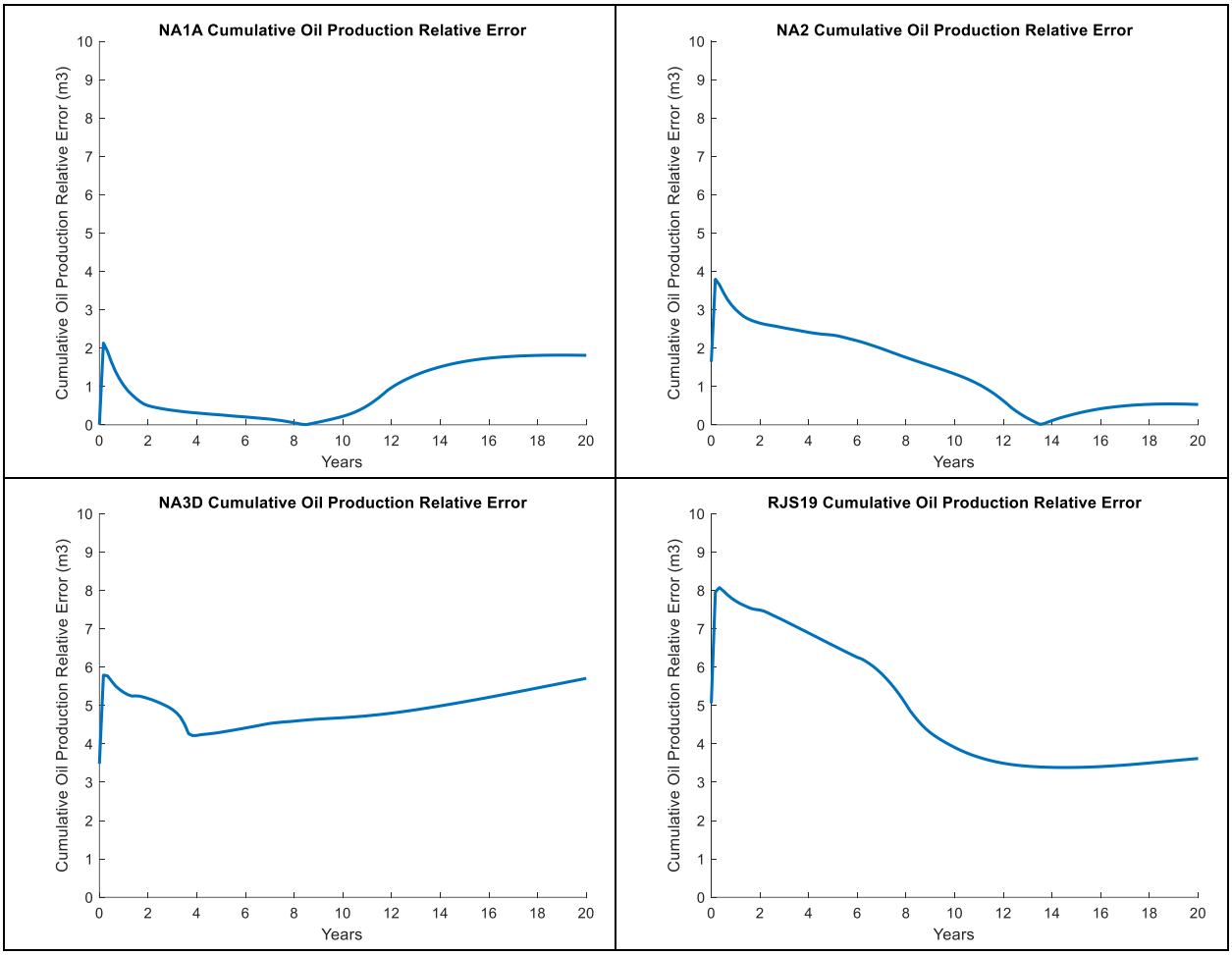


Figure 3.21 120 timesteps * 2 months cumulative oil production volume relative error (without validation).

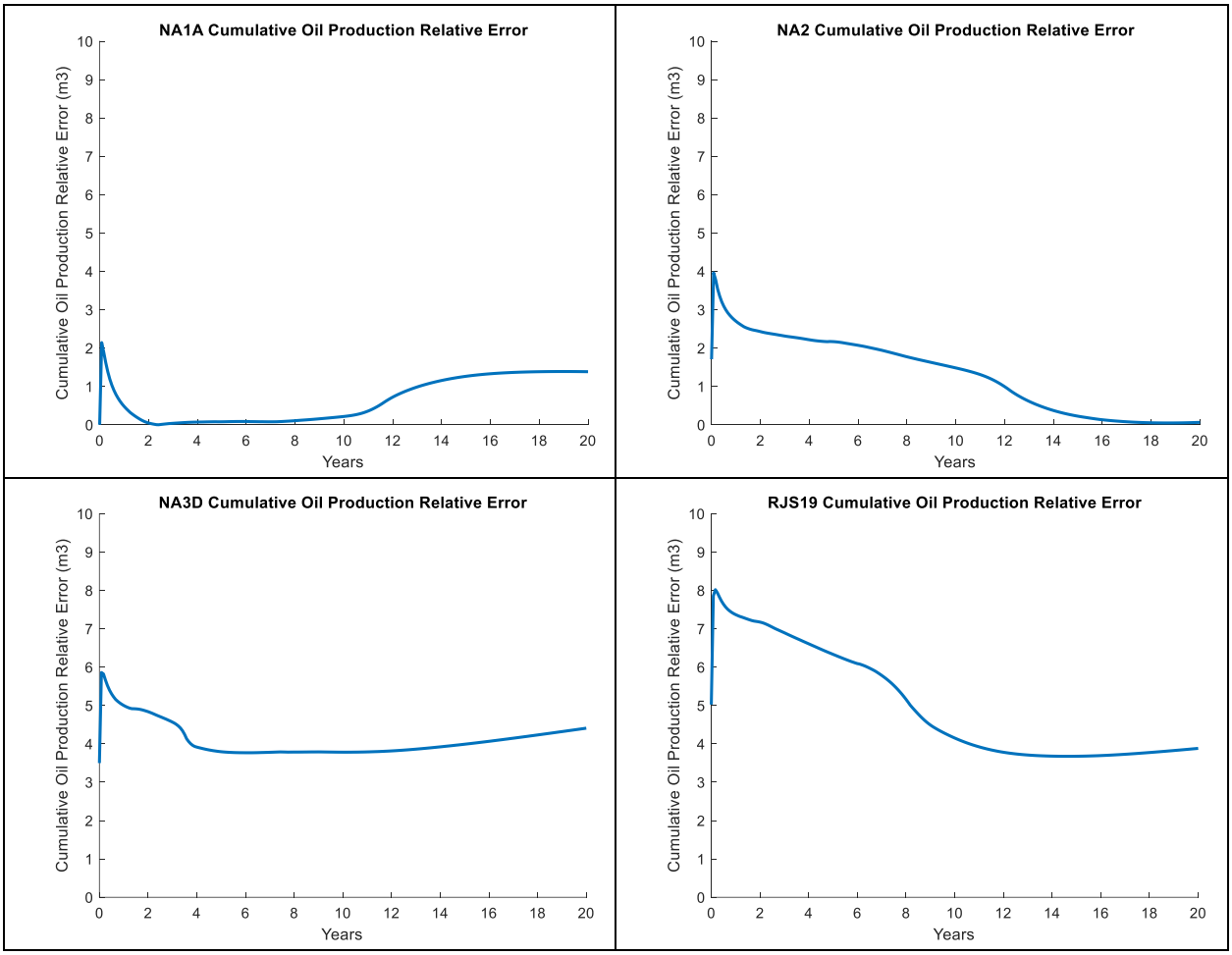


Figure 3.22 240 timesteps * 1 month cumulative oil production volume relative error (without validation).

3.1.1.4. Time reduction on UNISIM-I-D without validation

Table 3.3 presents the time data of implementing the NIROM method without using the validation method (step 4 in the chapter of Methodology). The proxy can reduce the computational time of running the high-fidelity simulation model significantly in all three cases. In the timestep selection that has generated the most accurate prediction results of the states using the most timesteps in the training set (240 steps * 1 month = 20 years), the proxy can reduce 60.65% of the computational time for each round of running the model compared with running the high-fidelity model.

When larger number of timesteps are used for training, the time reduction ratio decreases. The reason for this is that when more snapshots are used for training, more weighting factors are needed for building the surrogate model, because the weighting factor is corresponding to the distance between the reduced snapshots of each two timesteps. Therefore, in the process of prediction, more computation is needed when iteratively computing the reduced snapshot of the next timestep with the higher dimensional surrogate model.

Therefore, we can expect that if too many snapshots are used for training, the time reduced ratio will eventually decrease to zero or even negative. The balance between the accuracy of prediction and the computational time needs to be considered in most of the model reduction problems.

Table 3.3 Time data of the NIROM method for UNISIM-I-D without validation.

Timesteps	High-Fidelity Commercial Simulator (CMG) Running Time (s)	Training Time (s)	Testing Time (s)	Computational Time Reduction Ratio
60 steps * 4 months = 20 years	79.69	34.52	6.17	92.26%
120 steps * 2 months = 20 years	124.9	100.63	19.49	84.40%
240 steps * 1 month = 20 years	203.38	379.88	80.03	60.65%

3.1.2. NIROM method on UNISIM-I-D model with validation

In this section, the results of implementing the algorithm of improving the performance of the NIROM method when multiple variation trends of the state exist are presented. A validation set is needed for identifying the possible turning points of the variation trend of the state. The flowrate setup of the validation set is shown in **Table 3.4**. Except the additional validation set, all the other setup of the model (e.g. flowrates for setting up the training and testing set and the POD orders) are the same as before.

Table 3.4 The initial oil flowrates for the training set, the validation set, and the testing set for each well in the case study of UNISIM-I-D.

Well Name	Training Set Initial Oil Production Rate (m3/d)					Validation Set Initial Oil Production Rate (m3/d)	Testing Set Initial Oil Production Rate (m3/d)
NA1A	900	950	1000	1050	1100	925	925
NA2	900	950	1000	1050	1100	925	975
NA3D	900	950	1000	1050	1100	925	1025
RJS19	900	950	1000	1050	1100	925	1075

3.1.2.1. Pressure matching results on UNISIM-I-D with validation

Figure 3.23 shows the pressure matching results for the three different timestep selections after implementing the validation algorithm. **Figure 3.24** and **Figure 3.25** show the average relative error of the pressure predicted by the proxy in normal grids and well grids respectively. The shortest timestep interval has the best accuracy results (within 6.25% for normal grids and 7.46% for well grids), and the longest timestep interval has the worst results (within 11.68 % for normal grids and 13.95% for well grids). Compared with **Figure 3.2** and **Figure 3.3**, the proxy using the validation algorithm has better performance in the well grids and slightly worse accuracy in the normal grids.

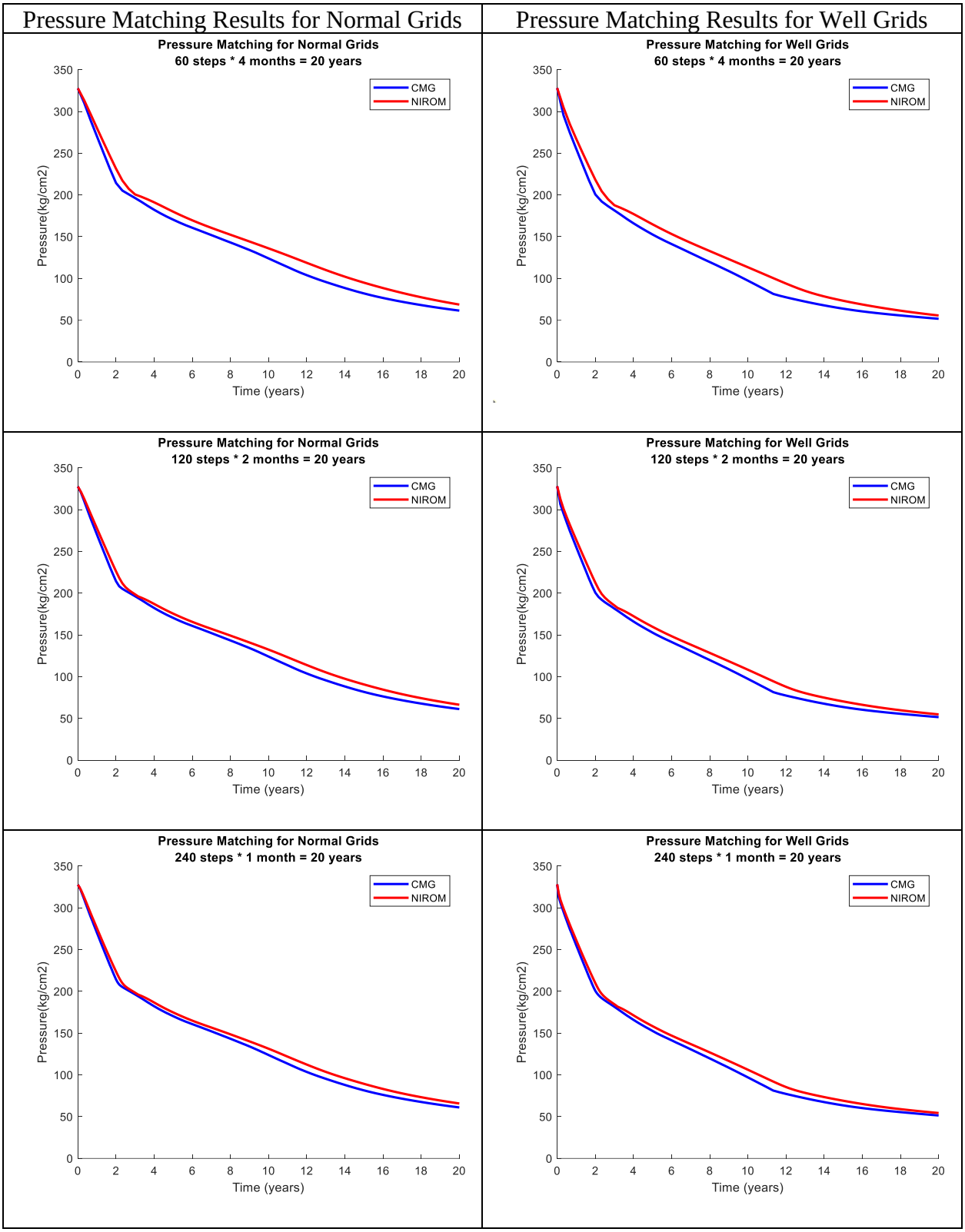


Figure 3.23 Pressure matching results of 3 different selections of timestep interval for UNISIM-I-D with validation.

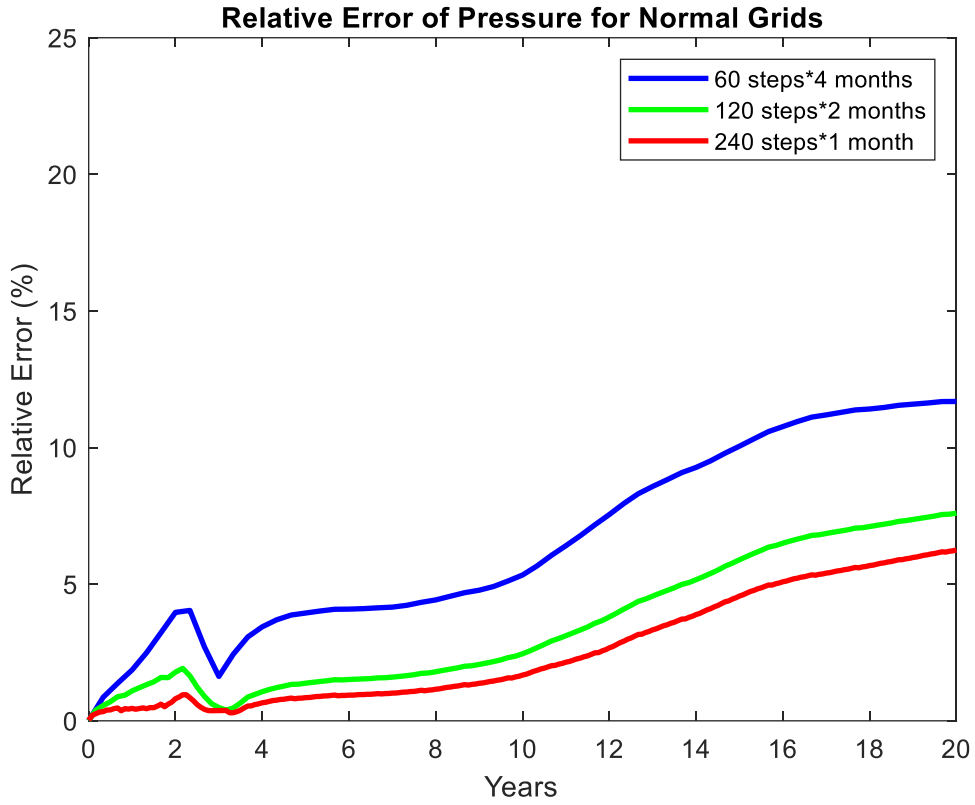


Figure 3.24 Average relative error of pressure for normal grids in UNISIM-I-D with validation.

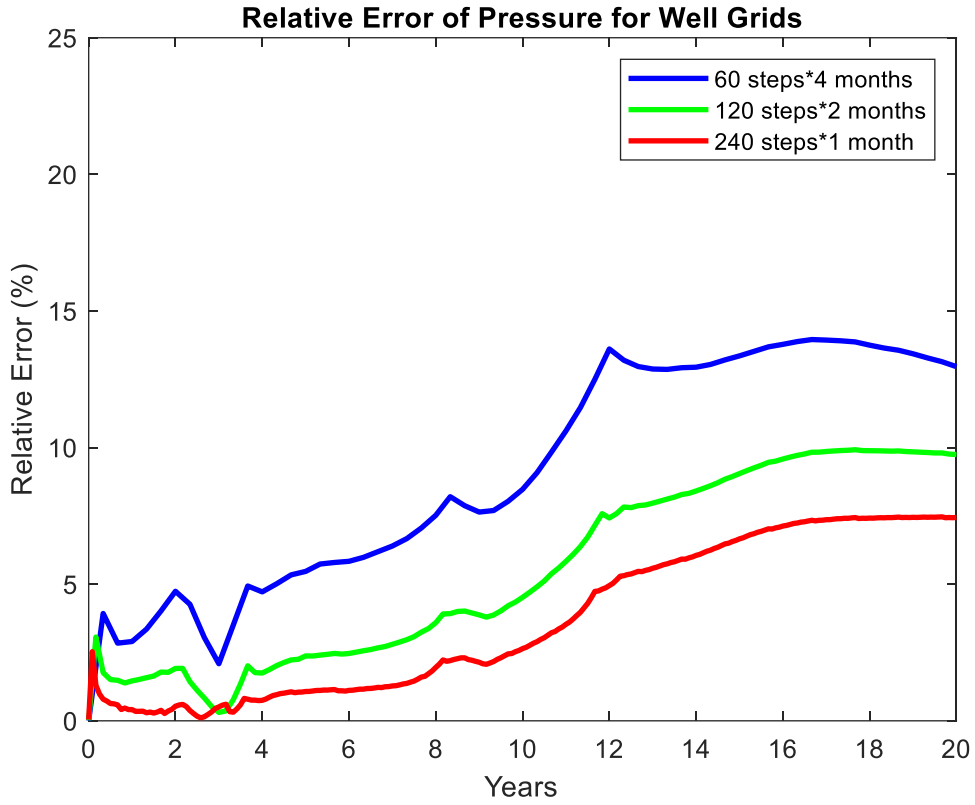


Figure 3.25 Average relative error of pressure for well grids in UNISIM-I-D with validation.

The left column of **Figure 3.26** shows the pressure map obtained by the proxy for the three different timestep selections after the validation algorithm is implemented, and the right column shows the corresponding relative error map. Compared with **Figure 3.5**, the validation algorithm does not change the results obviously, because the variation of pressure is relative simpler than the variation of oil saturation, and the pressure prediction results for the proxy before is already quite accurate.

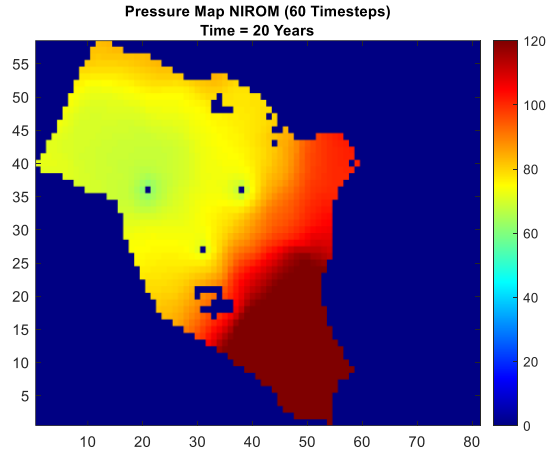
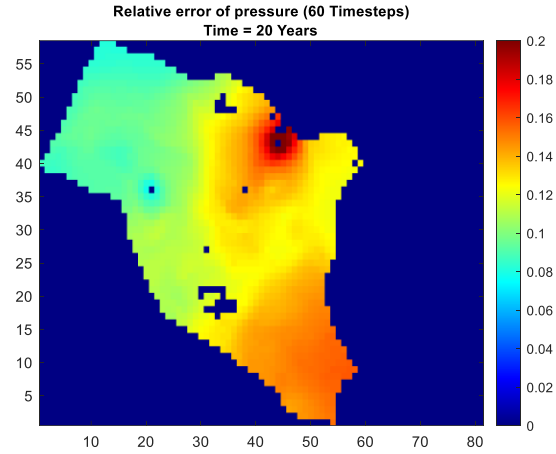
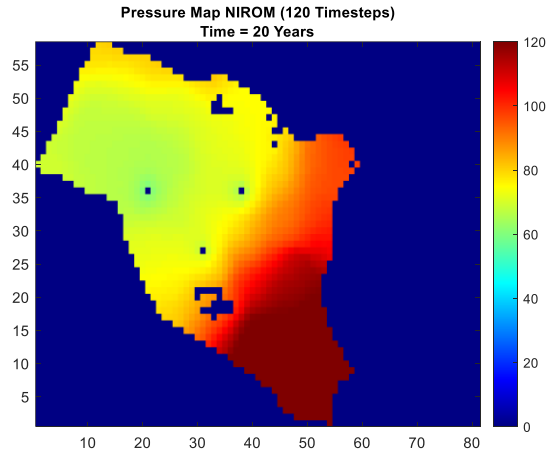
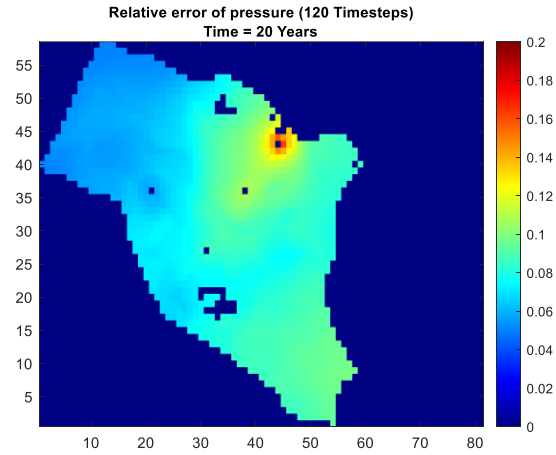
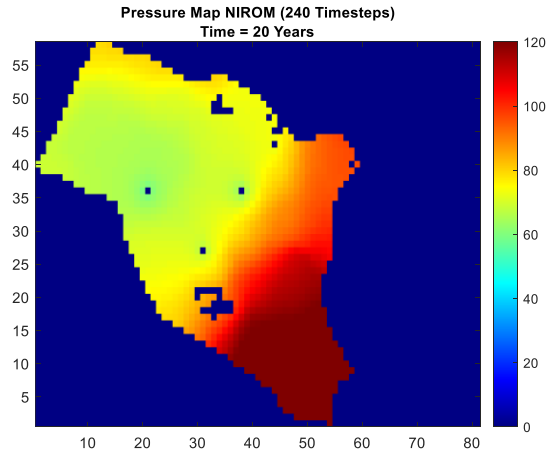
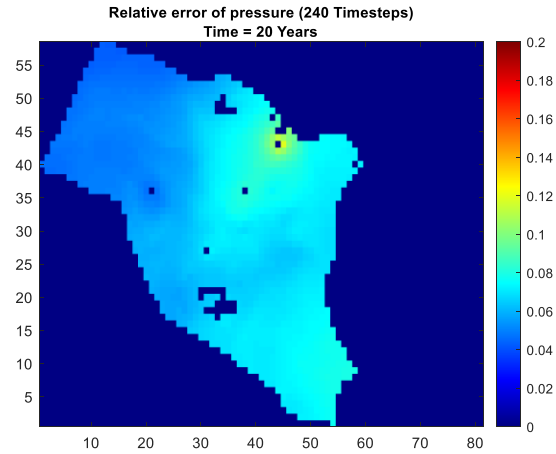
Pressure Map of the First Layer of UNISIM-I-D Obtained by the NIROM Method After Implementing the Validation Algorithm	Relative Error of Pressure in the First Layer of UNISIM-I-D After Implementing the Validation Algorithm
	
	
	

Figure 3.26 Pressure map of layer 1 of UNISIM-I-D at 20 years obtained by the NIROM method with validation for 3 different selections of timestep intervals(left column). The relative error of pressure for each grid (right column)

3.1.2.2. Oil saturation matching results on UNISIM-I-D with validation

Figure 3.27 shows the oil saturation matching results for the three different timestep selections after implementing the validation algorithm. Compared with **Figure 3.6**, the oil saturation curve obtained by the NIROM method is much closer to the curve generated by the high-fidelity model in both normal grids and well grids. The performance of the proxy in predicting multiple variation behaviors of the oil saturation is significantly improved, and this time, the proxy can capture the residue oil saturation correctly, which is severely over-estimated when the algorithm is not implemented. The proxy is still not able to perfectly capture the turning points even it is built up with the consider of multiple variation trends of the state due to the nature of RBF algorithm. However, this issue can be improved by using shorter timestep interval for the training set.

Figure 3.28 and **Figure 3.29** show the average relative error of oil saturation for normal grids and well grids after the validation algorithm is implemented respectively. The relative error clearly proves that the validation process has improved the accuracy of the NIROM method. Before the validation algorithm is implemented, the highest average relative error for the normal grids are 11% and 6% for the worst and best selections of timestep interval, which reduce to 4.76% and 4.13%. For the well grids, the highest average relative error reduces from 8.5% and 5.45% to 4.72% and 2.3% for the worst and best timestep selections.

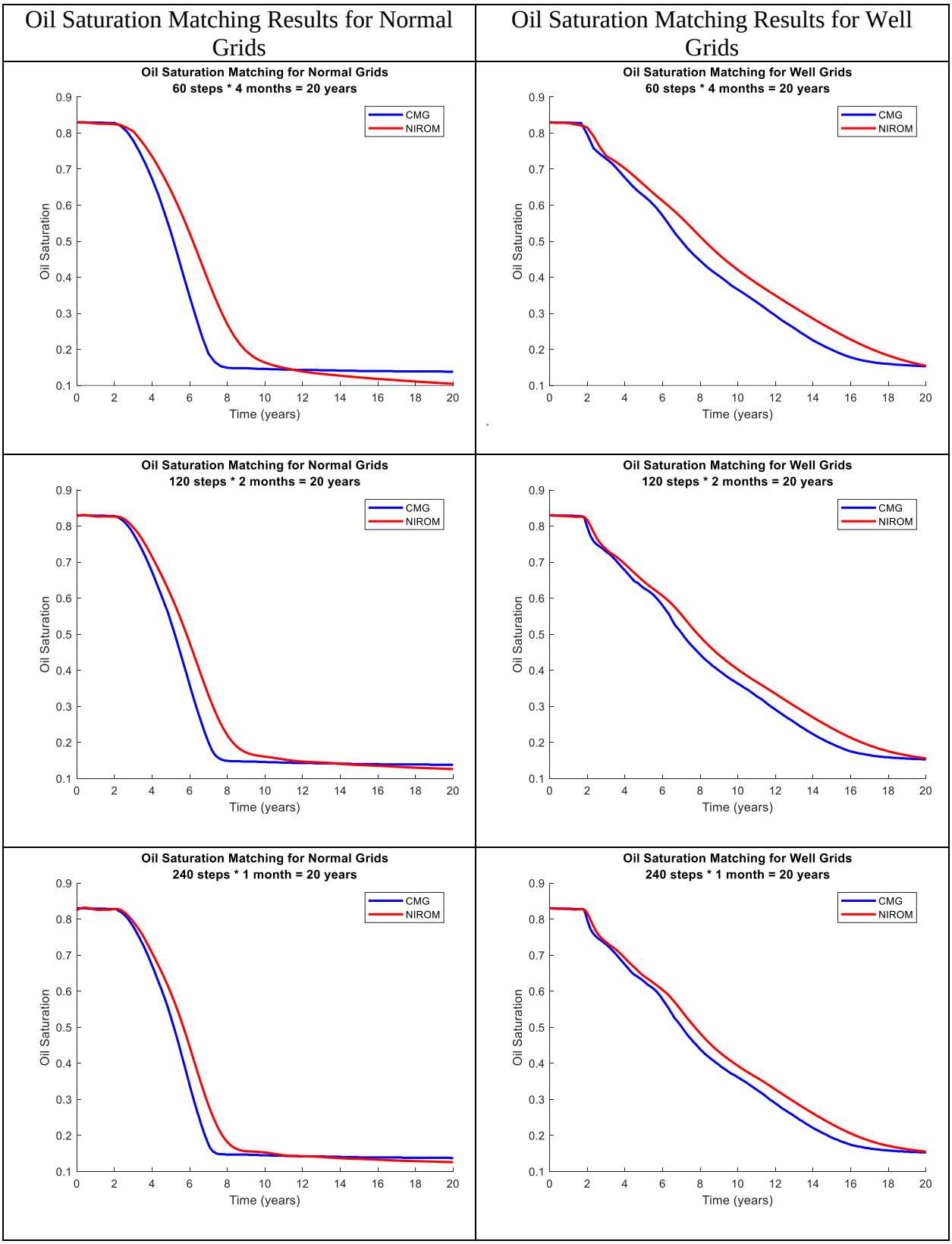


Figure 3.27 Oil saturation matching results of 3 different selections of timestep interval for UNISIM-I-D with validation.

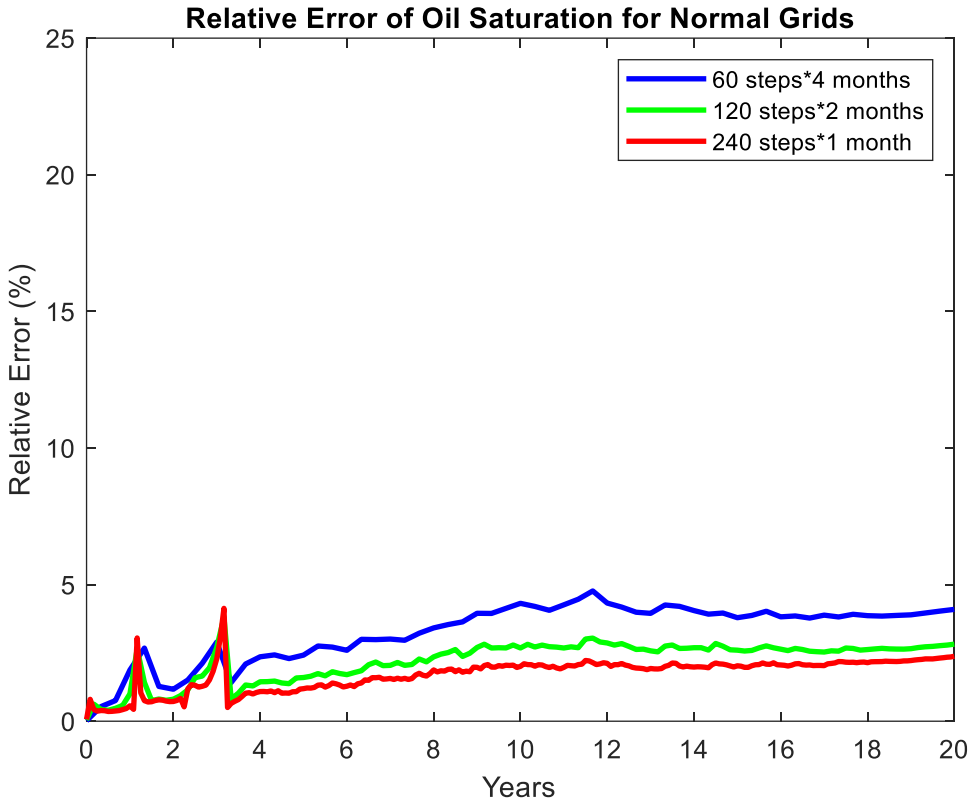


Figure 3.28 Average relative error of oil saturation for normal grids in UNISIM-I-D with validation.

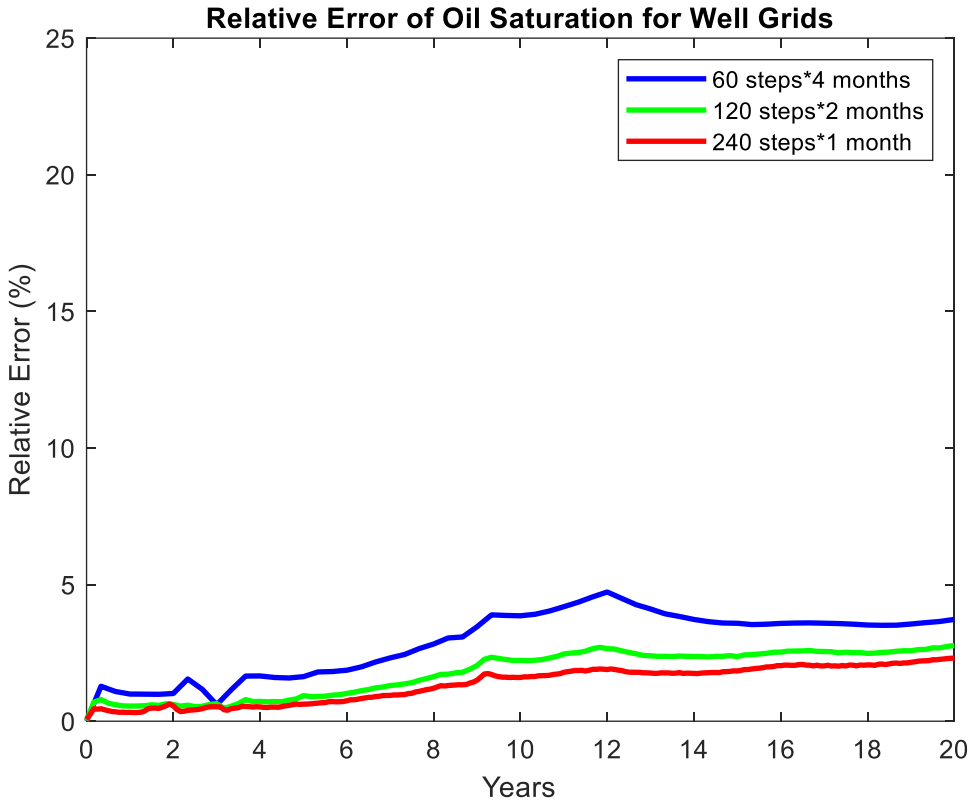


Figure 3.29 Average relative error of oil saturation for well grids in UNISIM-I-D with validation.

The left column of Figure 3.30 shows the oil saturation map predicted by the proxy for the three different timestep selections after the validation algorithm is implemented, and the right column shows the corresponding relative error map. Compared with **Figure 3.10**, the accuracy of the NIROM method is significantly improved, especially in the grids that have high porosity and permeability. This is because the NIROM method can better capture the relative sharper changes after the validation algorithm is implemented.

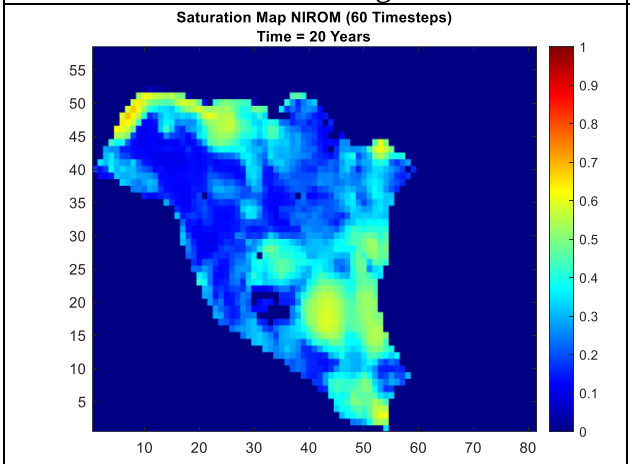
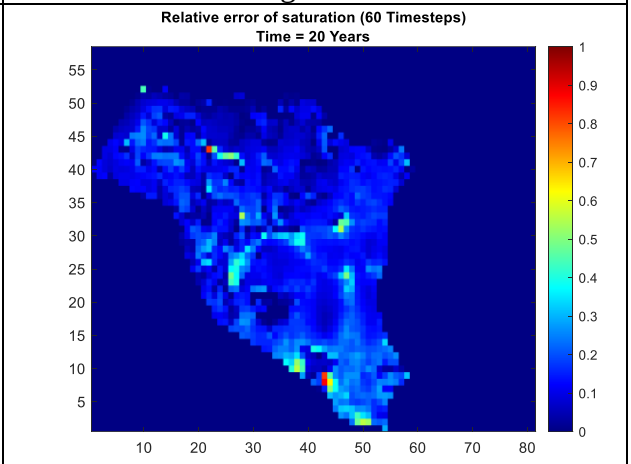
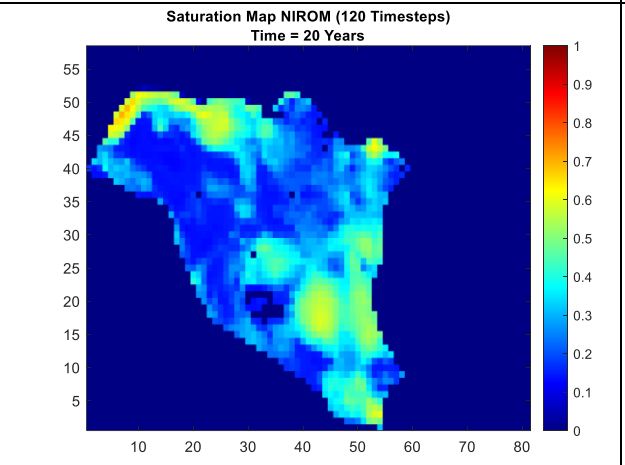
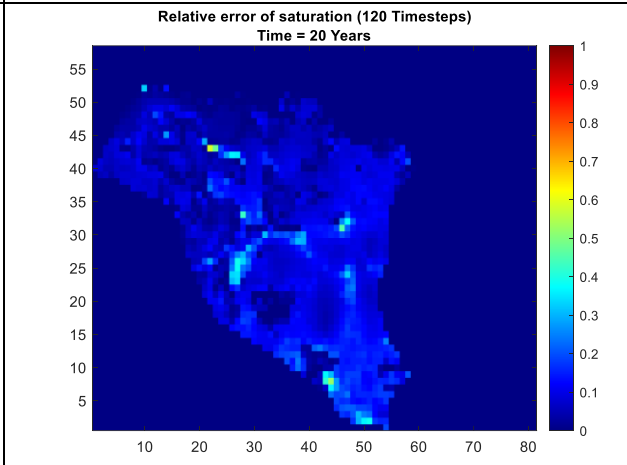
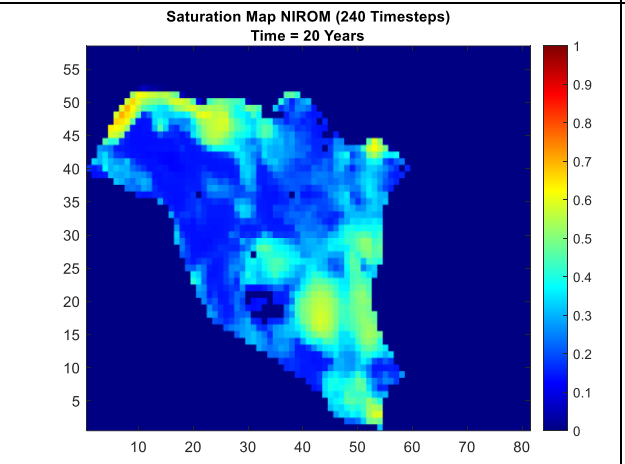
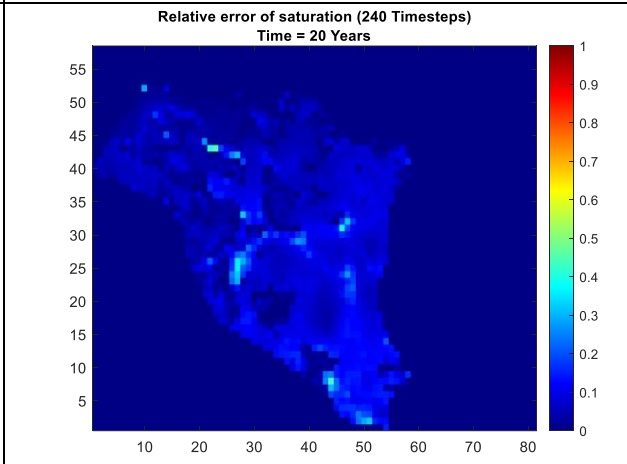
Oil Saturation Map of UNISIM-I-D Obtained by the NIROM Method After Implementing the Validation Algorithm	Relative Error of Oil Saturation of UNISIM-I-D After Implementing the Validation Algorithm
	
	
	

Figure 3.30 Oil saturation map of layer 1 of UNISIM-I-D at 20 years obtained by the NIROM method with validation for 3 different selections of timestep intervals(left column). The relative error of oil saturation for each grid (right column).

3.1.2.3. Oil rate matching results with validation

Figure 3.31 to **Figure 3.33** show the oil rate matching results for each producer when a validation data set is used to tune the surrogate model and make it capture the multiple variation trends of the state variables better. Compared with **Figure 3.11** to **Figure 3.13**, the NIROM method has generated much more accurate prediction at the switching point from constant oil rate production to constant bottom-hole pressure production.

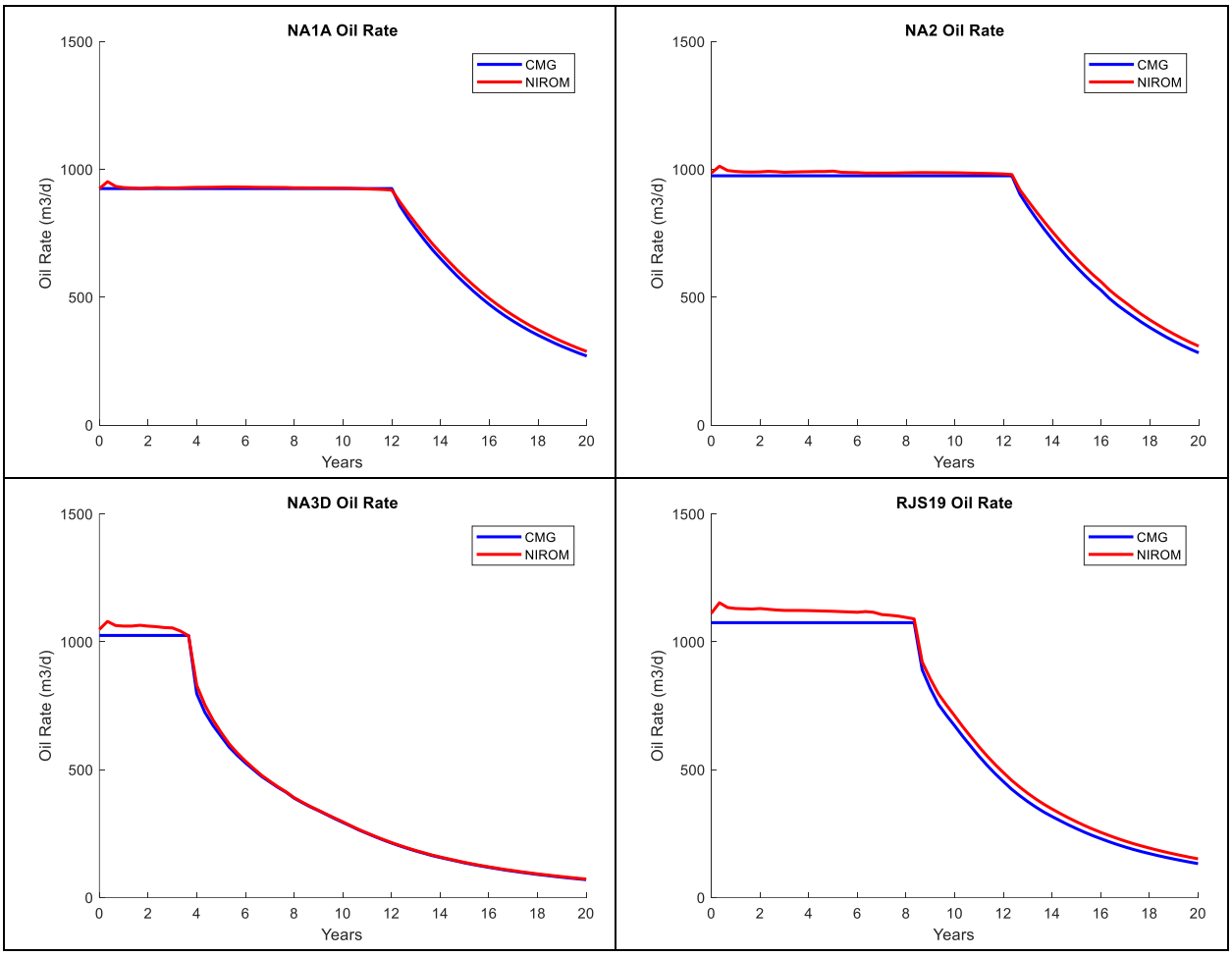


Figure 3.31 60 timesteps * 4 months oil rate matching results for each well (with validation).

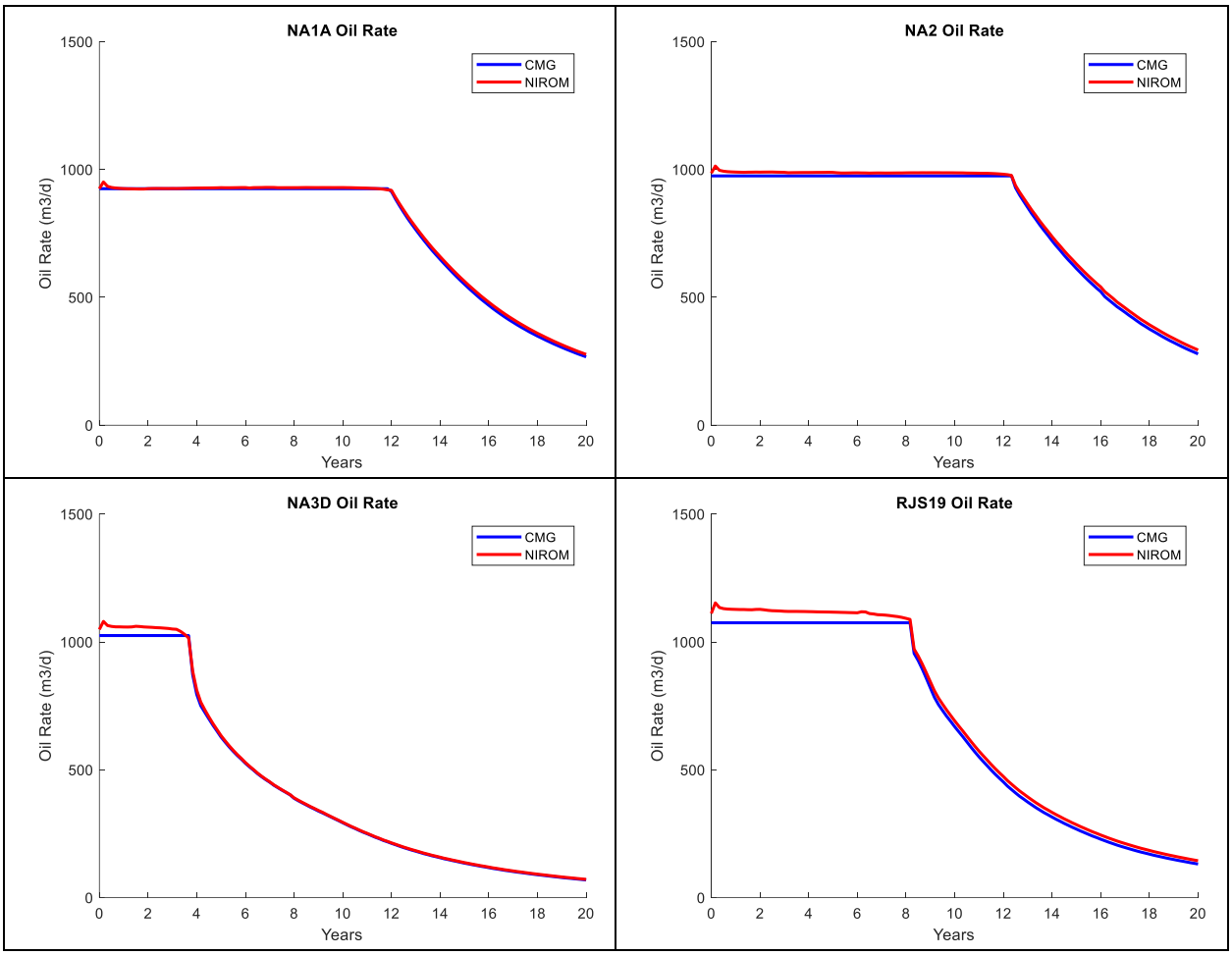


Figure 3.32 120 timesteps * 2 months oil rate matching results for each well (with validation).

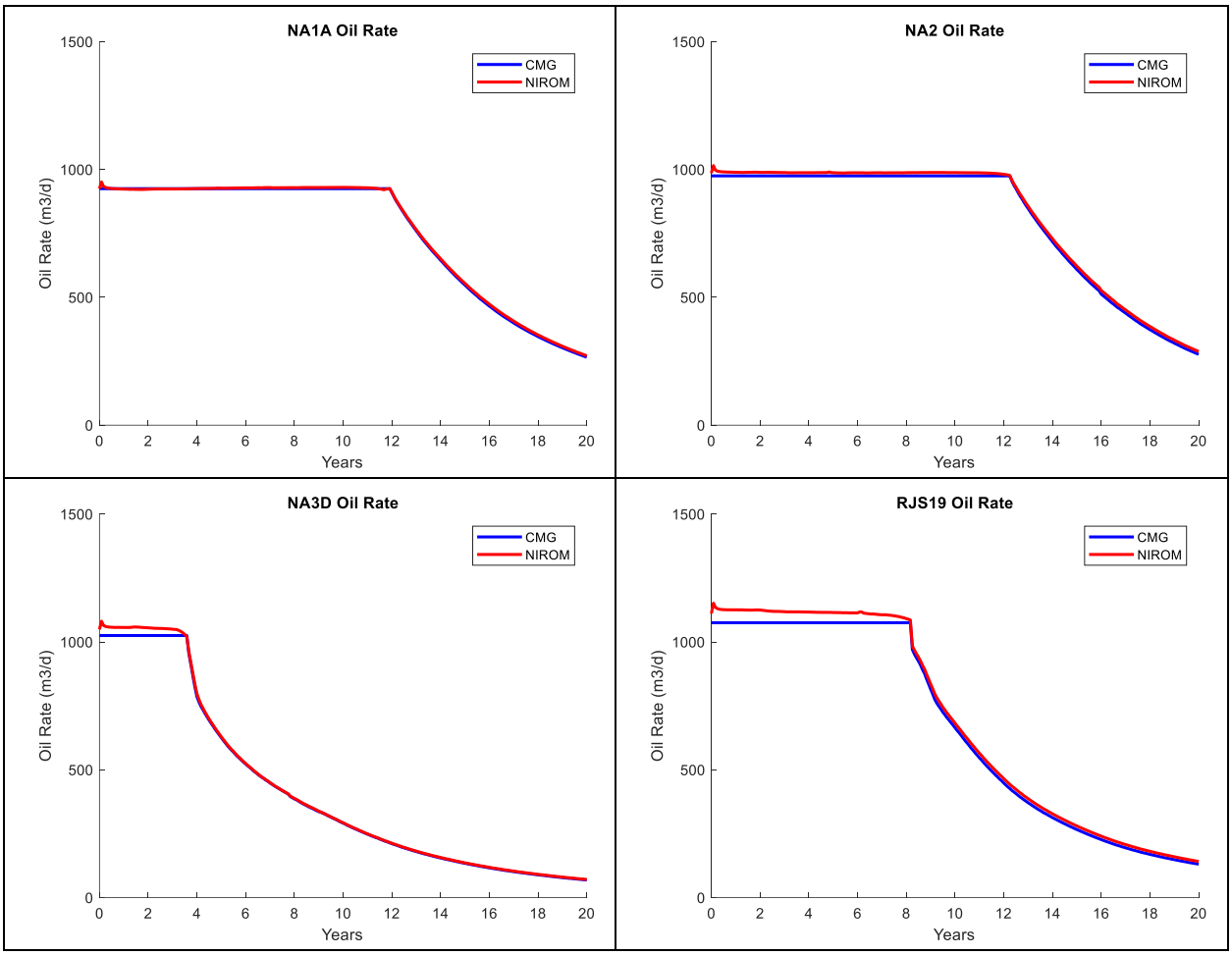


Figure 3.33 240 timesteps * 1 month oil rate matching results for each well (with validation).

Figure 3.34 to **Figure 3.36** present the relative error of oil production rate of the prediction results of the NIROM method. The case using 1 month as the timestep interval has generated the most accurate prediction. The highest relative error is 2.86%, 4.15%, 5.48%, 8.13% in well NA1A, NA2, NA3D, RJS19 respectively. Compared with the case without the validation process, all 4 wells have generated more accurate prediction in oil production rate.

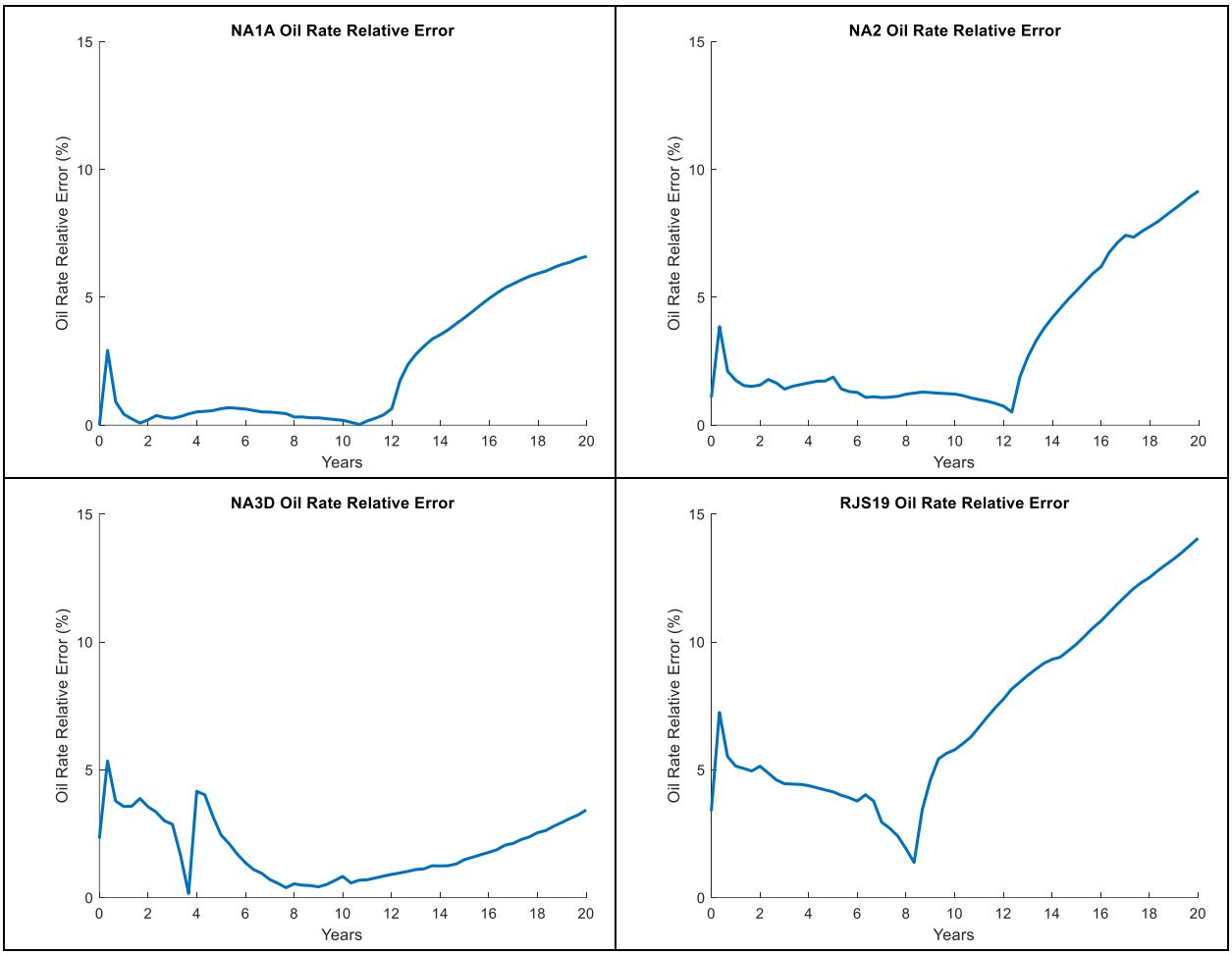


Figure 3.34 60 timesteps * 4 months oil rate relative error (with validation).

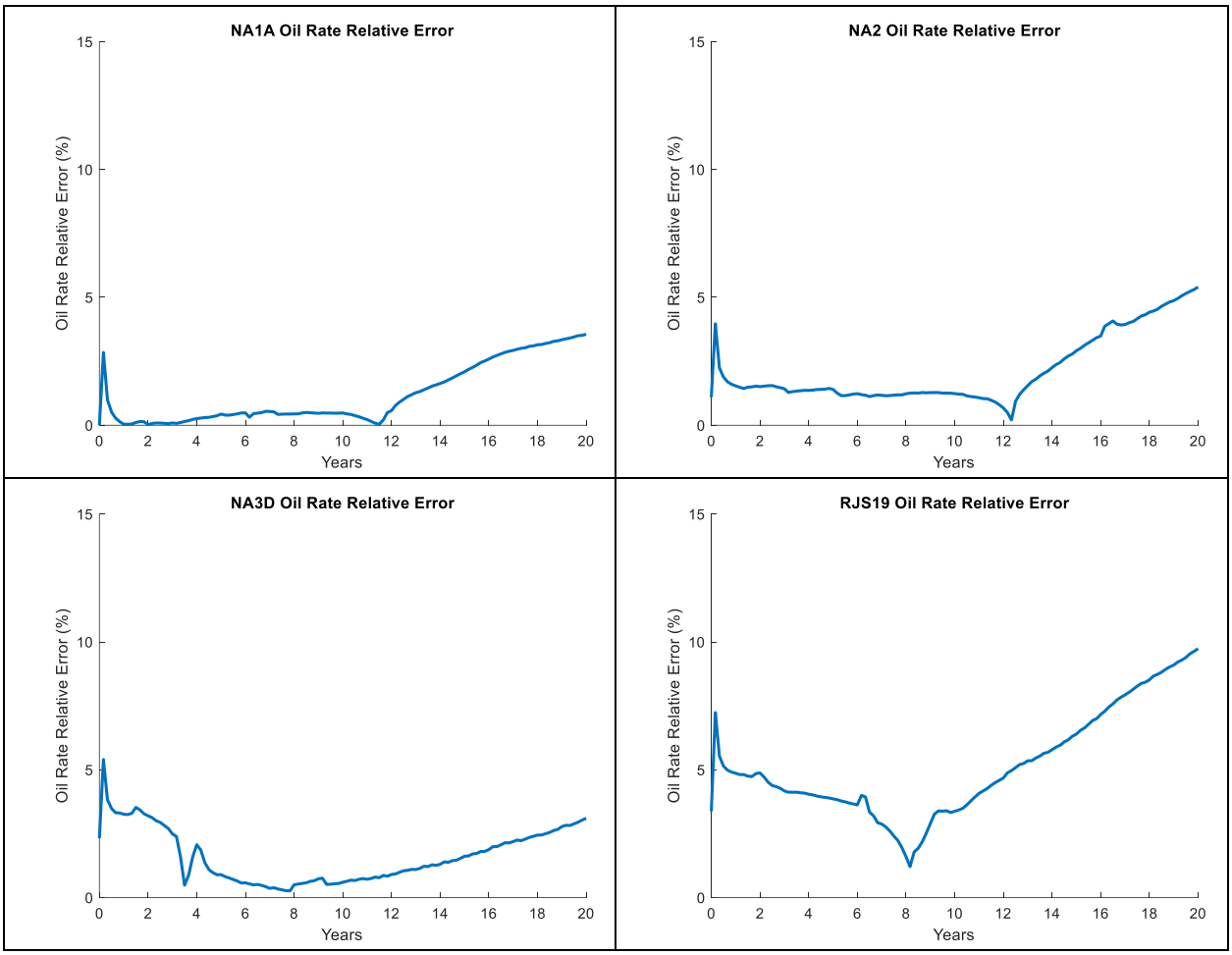


Figure 3.35 120 timesteps * 2 months oil rate relative error (with validation).

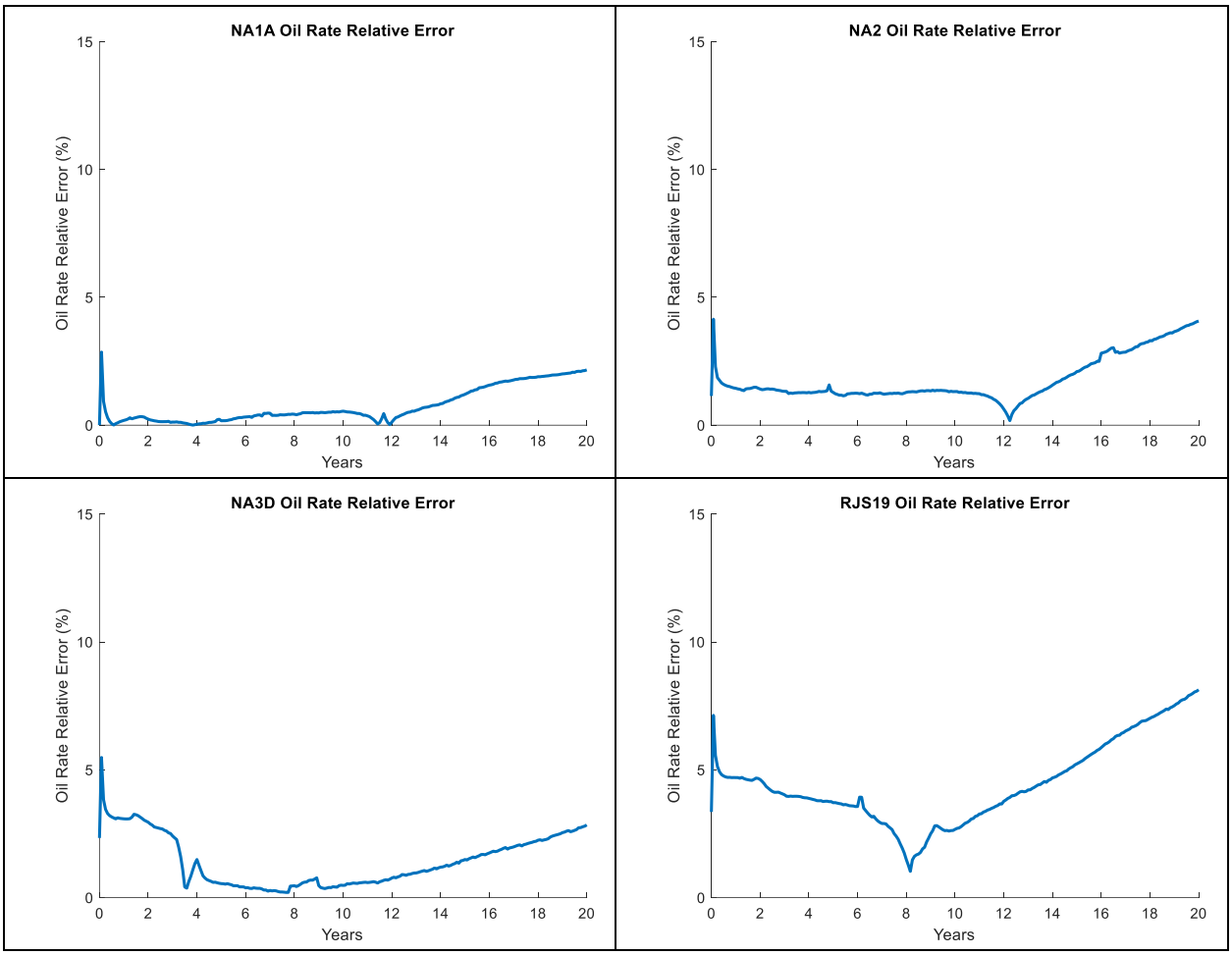


Figure 3.36 240 timesteps * 1 month oil rate relative error (with validation).

Figure 3.37 to **Figure 3.39** show the cumulative oil production volume prediction result in each well for each timestep case when the validation process is conducted. **Figure 3.40** to **Figure 3.42** show the correlated relative error. The case using 1 month as the timestep interval for training has generated the most accurate prediction results. The highest relative error in the most accurate prediction result is 1.43%, 2.64%, 3.91%, 5.35% in well NA1A, NA2, NA3D, RJS19 respectively. The NIROM method has less relative error in all 4 producers compared with the no validation cases.

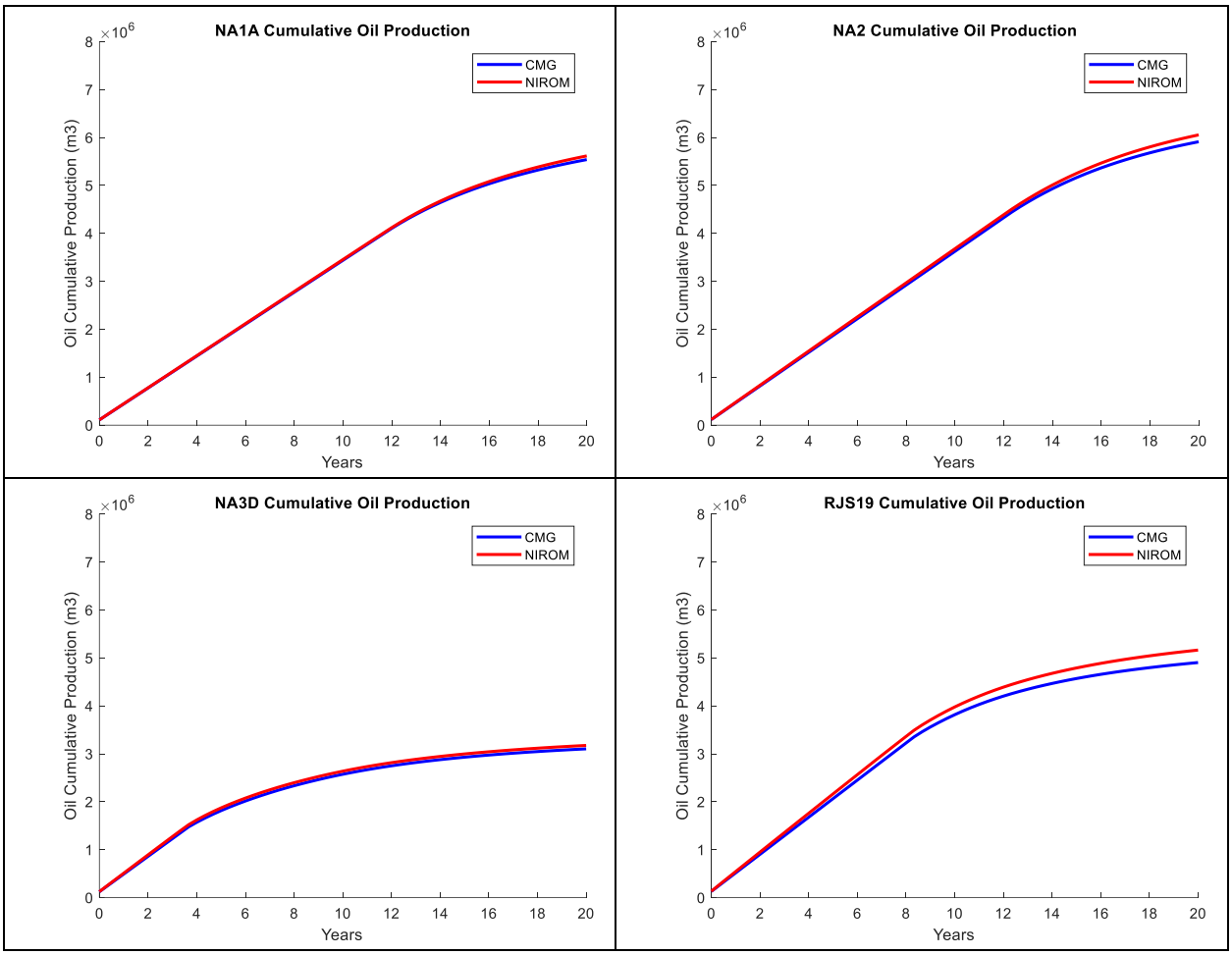


Figure 3.37 60 timesteps * 4 months cumulative oil production volume matching results for each well (with validation).

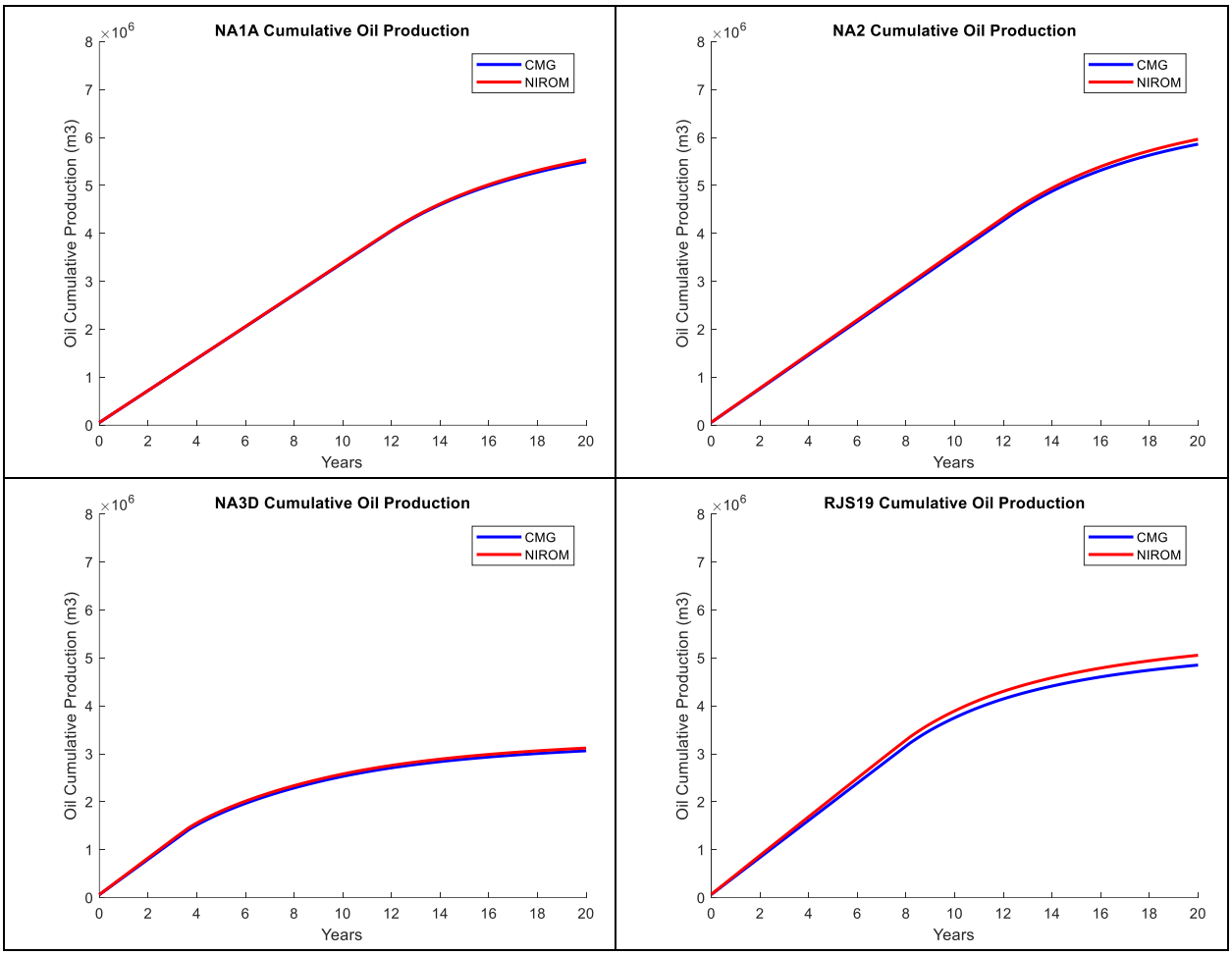


Figure 3.38 120 timesteps * 2 months cumulative oil production volume matching results for each well (with validation).

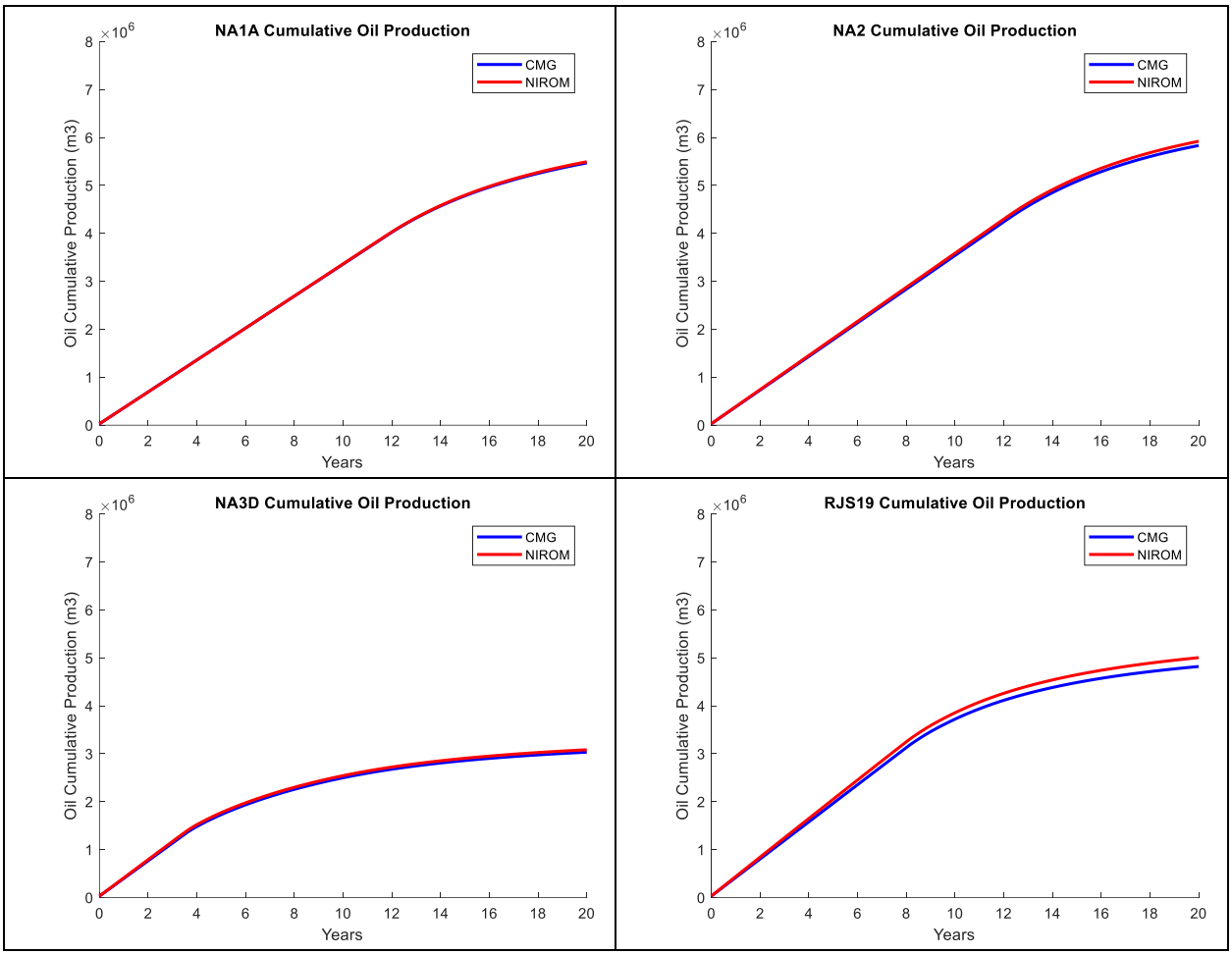


Figure 3.39 240 timesteps * 1 month cumulative oil production volume matching results for each well (with validation).

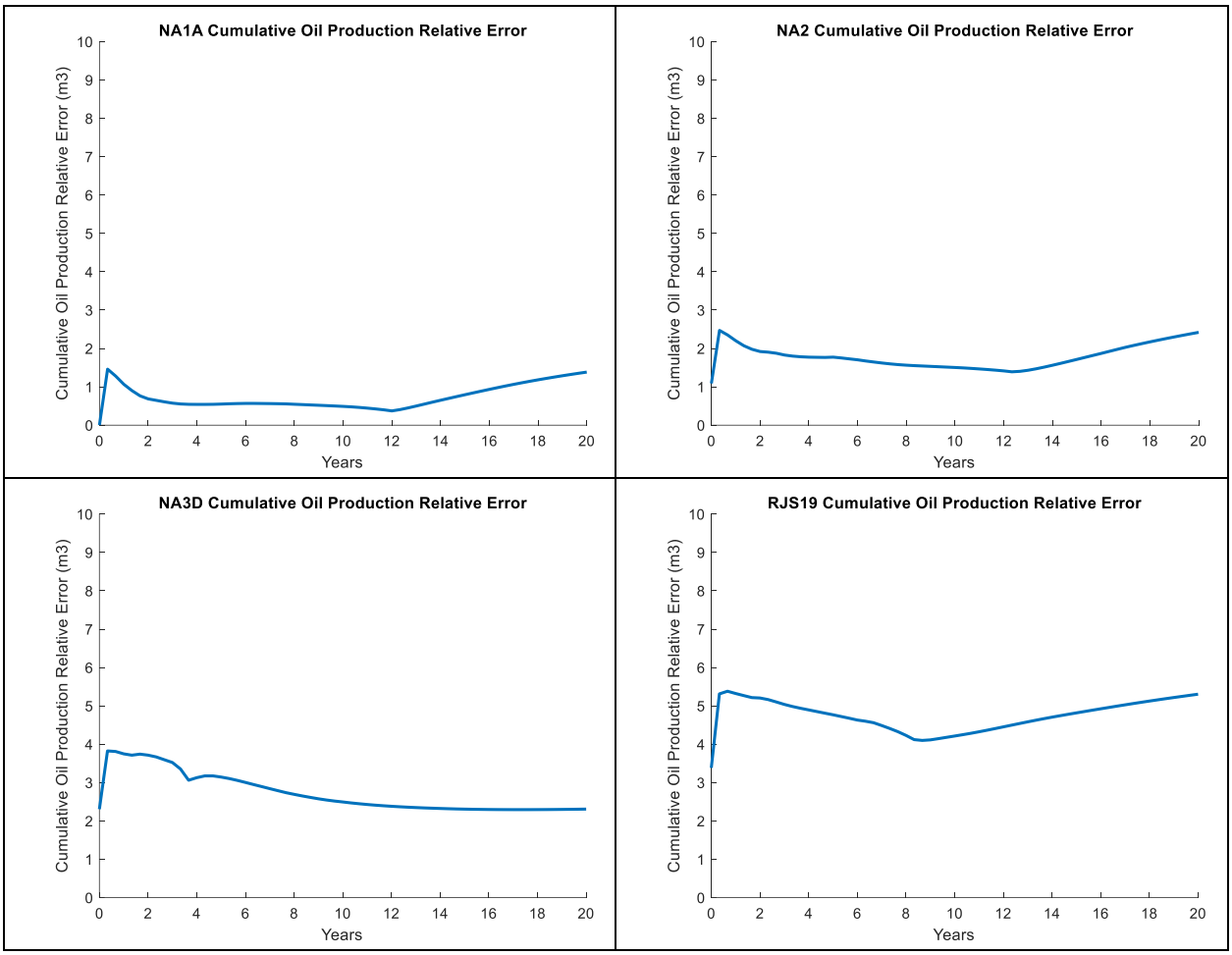


Figure 3.40 60 timesteps * 4 months cumulative oil production volume relative error (with validation).

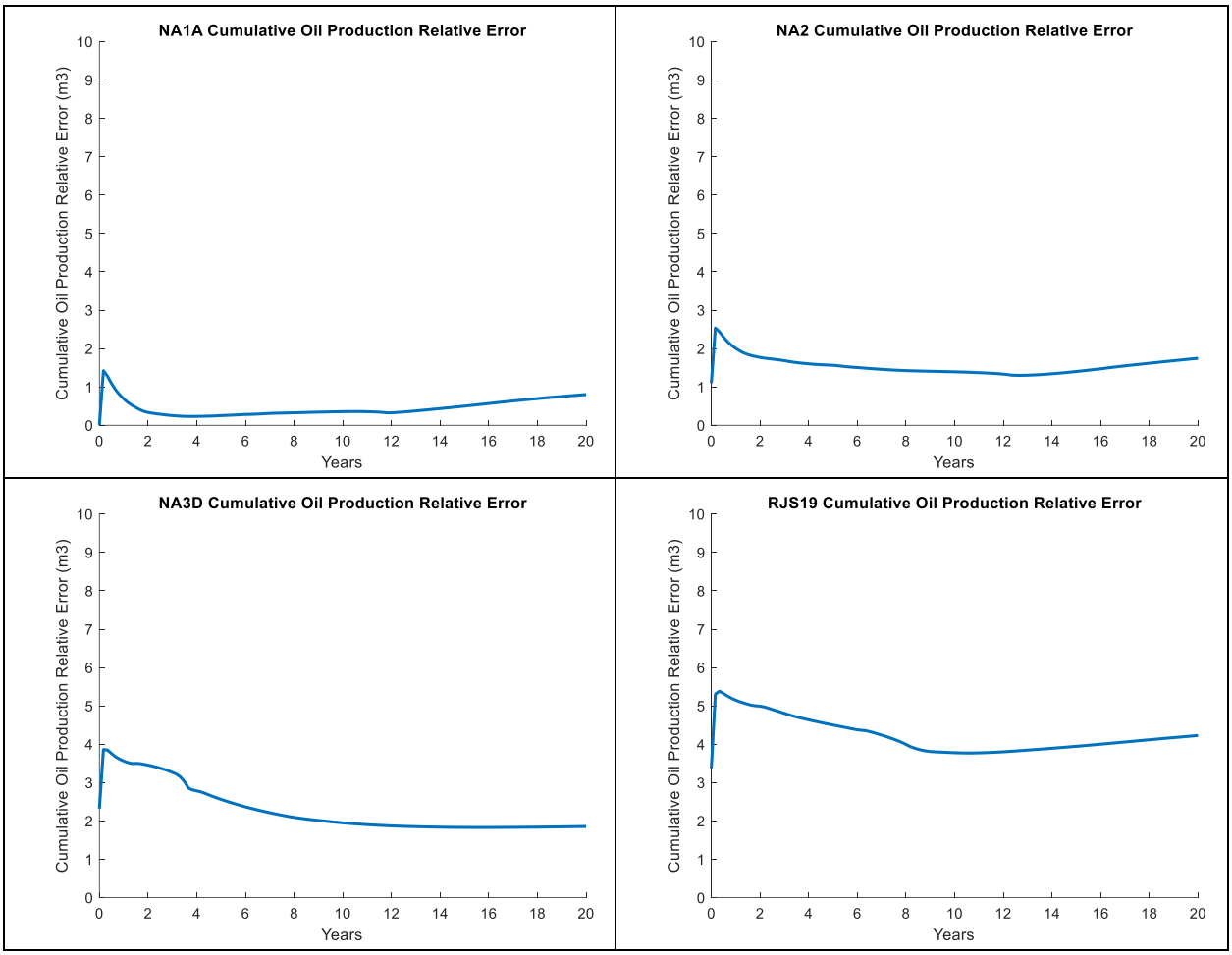


Figure 3.41 120 timesteps * 2 months cumulative oil production volume relative error (with validation).

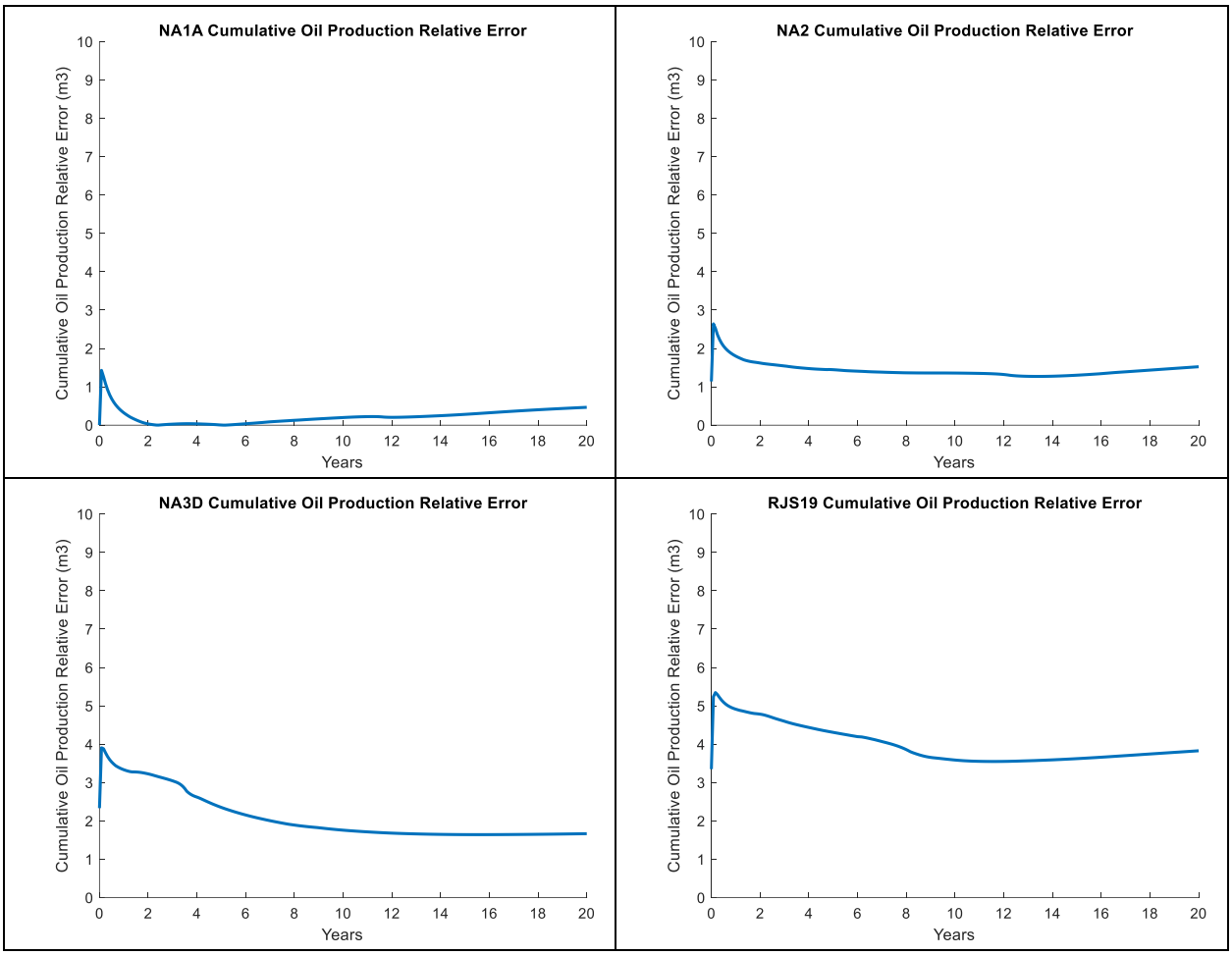


Figure 3.42 240 timesteps * 1 month cumulative oil production volume relative error (with validation).

3.1.2.4. Time reduction on UNISIM-I-D with validation

Table 3.5 presents the time data of implementing the NIROM method with the validation algorithm. Compared with **Table 3.5**, the training time significant increases, because more than one surrogate model are trained to capture multiple variation trends of the state after validation. The more complex the variation of the state, the more surrogate models are needed. Also, one round of validation is needed to identify the turning points of the trends of the variation of the state, which is taking additional computational time. However, all the time increased is in the offline process, and for the online process (testing), the computational time remains roughly the same as

before. Therefore, the increase in the offline time will become negligible compared with the time saved when many rounds of prediction of different simulation setups are run.

Table 3.5 Time data of the NIROM method for UNISIM-I-D with validation.

Timesteps	High-Fidelity Commercial Simulator (CMG) Running Time (s)	Offline		Online	Computational Time Reduction Ratio
		Training Time (s)	Validation Time (s)	Testing Time (s)	
60 steps * 4 months = 20 years	79.69	54.51	5.49	5.26	93.40%
120 steps * 2 months = 20 years	124.9	181.84	19.33	18.68	85.04%
240 steps * 1 month = 20 years	203.38	714.15	81.38	73.18	64.02%

3.2. Geothermal Model

After discussing the performance of the NIROM method on UNISIM-I-D in which isothermal assumption is made, a geothermal model is built to assess the performance of the NIROM method when non-isothermal condition needs to be considered. In this model, 8 injection wells and 7 production wells are set up, and the location of the wells are shown in **Figure 1.2**. Each well is perforated from layer 3 to layer 12 and injecting or producing at constant water flowrate. The initial aquifer temperature is 170 ℃ and the injection wells are injecting water at 25 ℃.

The same as what we did in the UNISIM-I-D model, the results of not implementing the validation algorithm is shown firstly. Then, the results of implementing the NIROM method with validation are shown, and they are compared with the results of without validation to evaluate the effectiveness of the validation algorithm in identifying multiple variation trends of the state, and thus improve the performance of the NIROM method.

3.2.1. NIROM method on geothermal model without validation

The well injection rate and production rate used for the training dataset and testing dataset are shown in **Table 3.6**. The simulating time is around 5 years in this model. The Injection process

results in the increase of pressure near around the injection well for a very short piece of time (several hours). To capture this behavior correctly in the high-fidelity simulation model, a few short timesteps (4 hours) are used. In this case study, three different timestep cases are tested which are 8 steps * 4 hours + 60 steps * 30 days, 8 steps * 4 hours + 120 steps * 15 days, 8 steps * 4 hours + 360 steps * 5 days. The performance of the NIROM method in predicting the pressure and temperature with these three timestep selections are shown in the following sections. The POD orders used for temperature and pressure of normal grids, injection well grids and production well grids are shown in **Table 3.7**.

Table 3.6 The initial water flowrates for the training set and the testing set for each well in the case study of the geothermal model without validation.

	Training Set Well Injection/Production Rate (m3/d)					Testing Set Well Injection/Production Rate (m3/d)
Inj-1	625	675	725	775	825	650
Inj-2	625	675	725	775	825	650
Inj-3	625	675	725	775	825	650
Inj-4	625	675	725	775	825	650
Inj-5	625	675	725	775	825	650
Inj-6	625	675	725	775	825	650
Inj-7	625	675	725	775	825	650
Inj-8	625	675	725	775	825	650
Prod-1	2000	2100	2200	2300	2400	2350
Prod-2	2000	2100	2200	2300	2400	2350
Prod-3	2000	2100	2200	2300	2400	2350
Prod-4	2000	2100	2200	2300	2400	2350
Prod-5	2000	2100	2200	2300	2400	2350
Prod-6	2000	2100	2200	2300	2400	2350
Prod-7	2000	2100	2200	2300	2400	2350

Table 3.7 The POD order for each state variable used for the geothermal model.

	Normal Grids Pressure	Injection Well Grids Temperature	Production Well Grids Pressure	Normal Grids Temperature	Injection Well Grids Temperature	Production Well Grids Temperature
POD order	10	10	10	10	10	10

3.2.1.1. Pressure matching results on geothermal model without validation

Figure 3.43 presents the pressure matching results of normal grids of the three different timestep selections, and **Figure 3.44** shows the average relative error for pressure in normal grids of different timestep cases. The pressure decrease of the normal grids in the geothermal model is relative smoother than it of the UNISIM-I-D model. Therefore, the length of the timestep interval used for training does not affect the accuracy of the NIROM method as obvious as in UNISIM-I-D, and the average relative error of all three cases are within 9%.

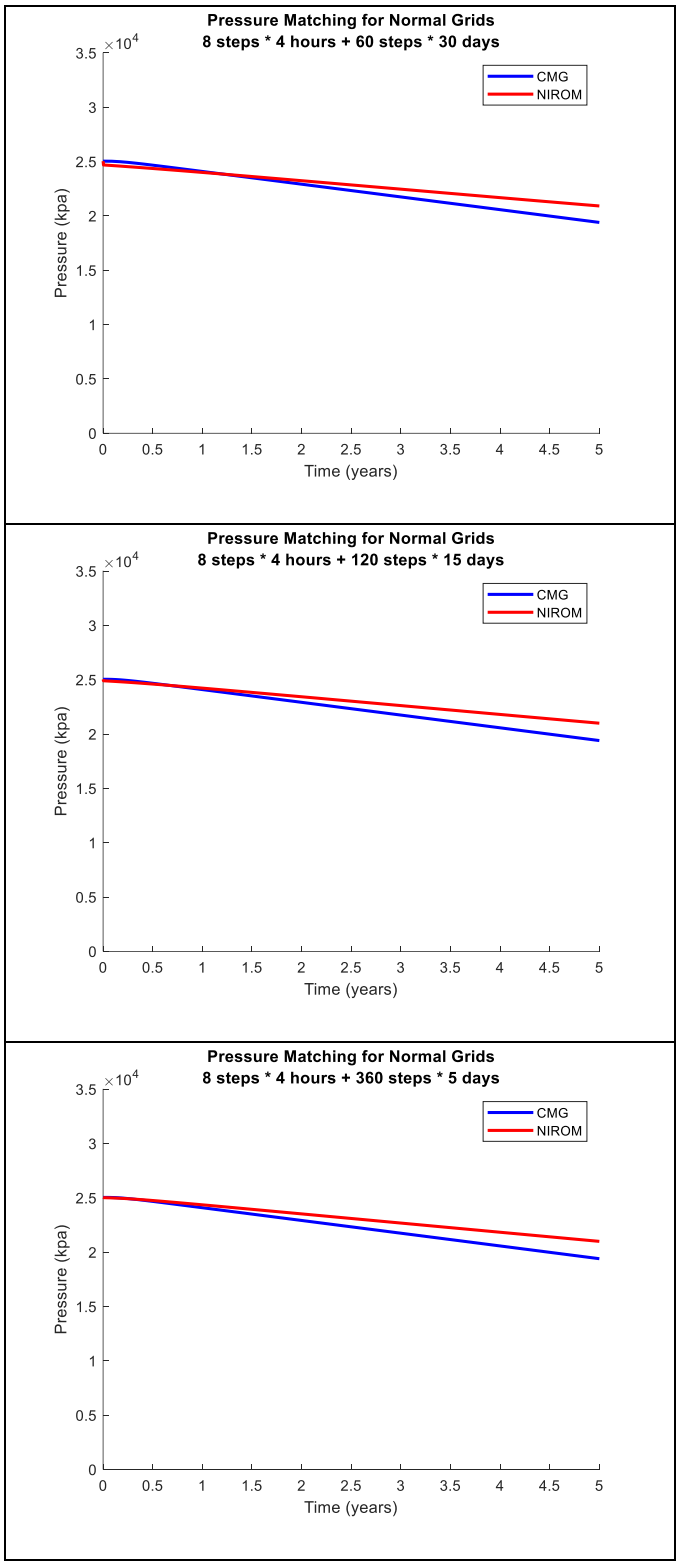


Figure 3.43 Pressure matching results for normal grids of 3 different selections of timestep interval for the geothermal model without validation.

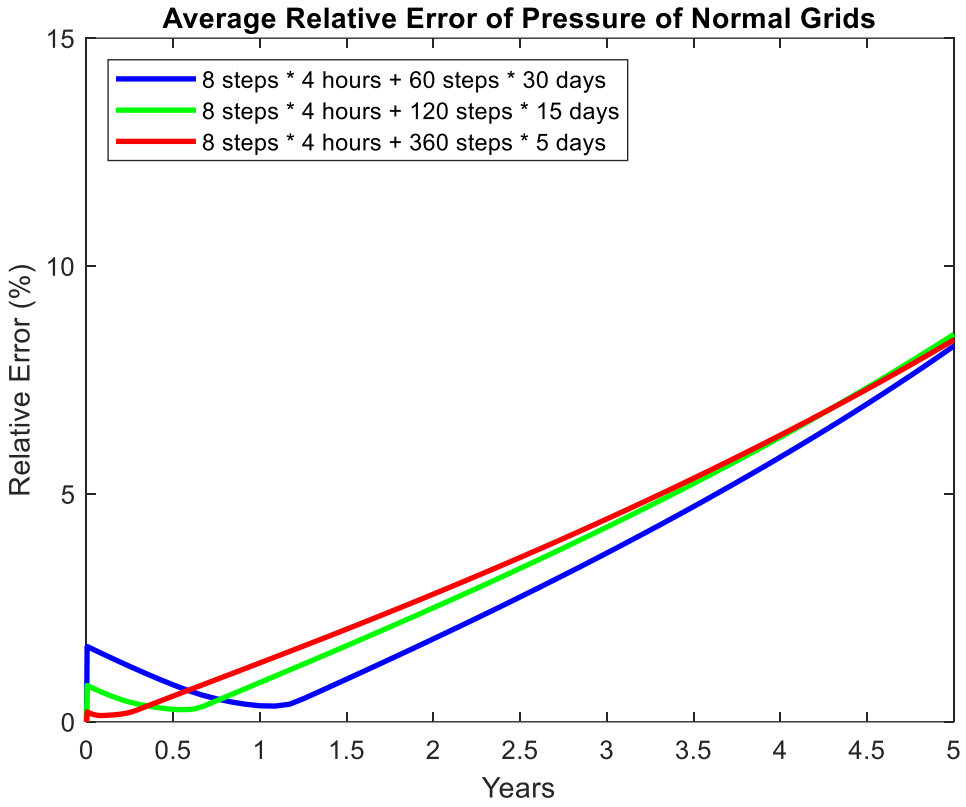


Figure 3.44 Average relative error of pressure for normal grids in the geothermal model without validation.

Figure 3.45 shows the Pressure matching results for the injection well and production well grids. For all the three cases, the average relative error is generally within 10%. Compared with the UNISIM-I-D model, the geothermal model has much simpler geological properties (homogeneous and isotropic) but more complex operating scheme (both injection and production wells are used). The proxy has relative worse pressure matching results in this geothermal case due to the injection wells. The injection wells cause the pressure increasing for a very short piece of time and this increasing variation trend of pressure is inverse to the general pressure decreasing trend, which is hard for the proxy to capture. Therefore, the prediction of the proxy is a little off at the very beginning of the simulation, and this is cumulating gradually to the end of the simulation.

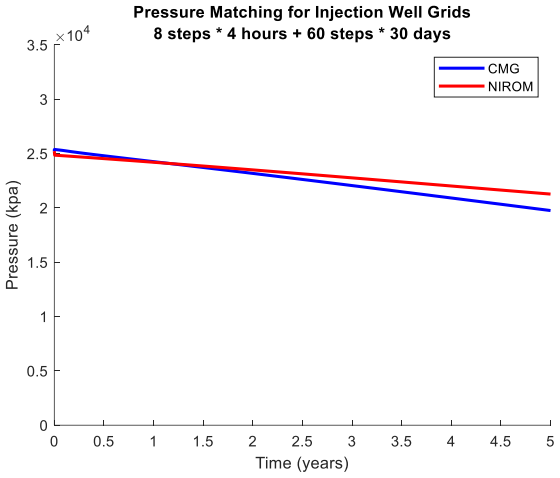
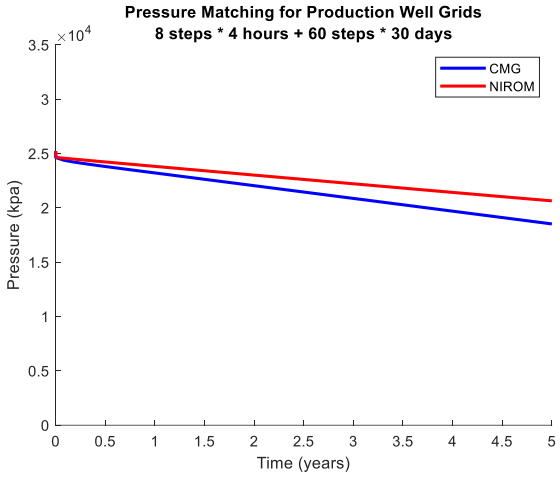
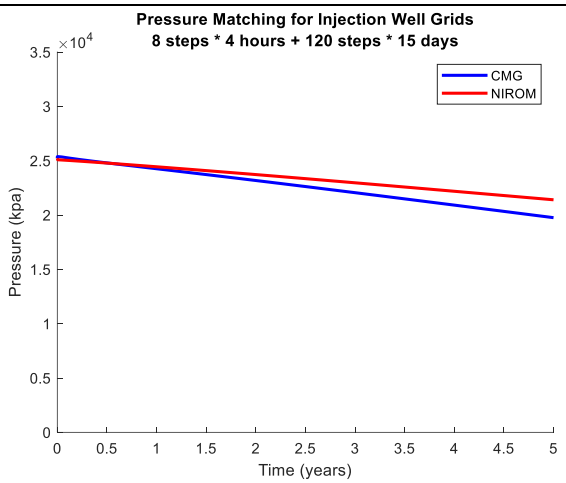
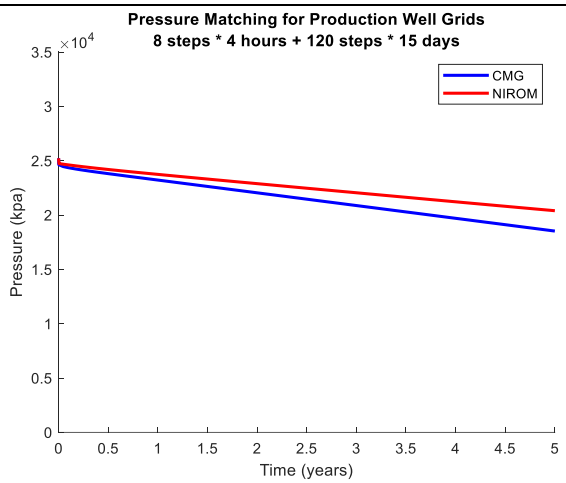
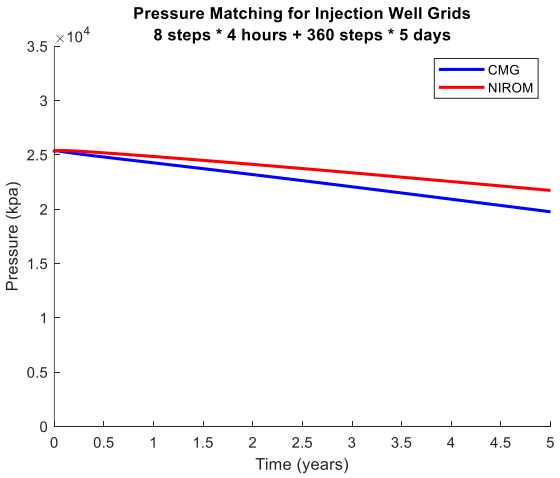
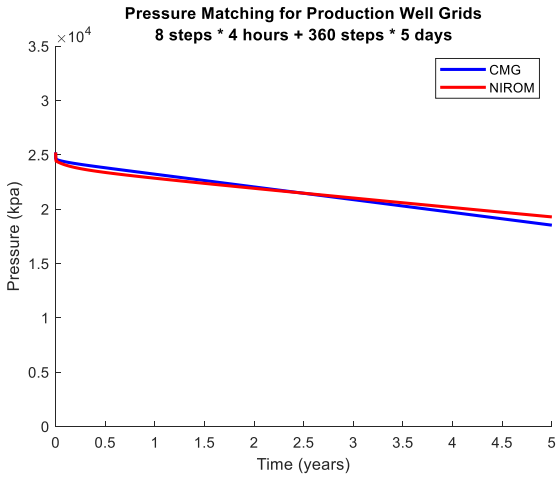
Pressure Matching Results for Injection Well Grids	Pressure Matching Results for Production Well Grids
	
	
	

Figure 3.45 Pressure matching results for well grids of 3 different selections of timestep interval for the geothermal model without validation.

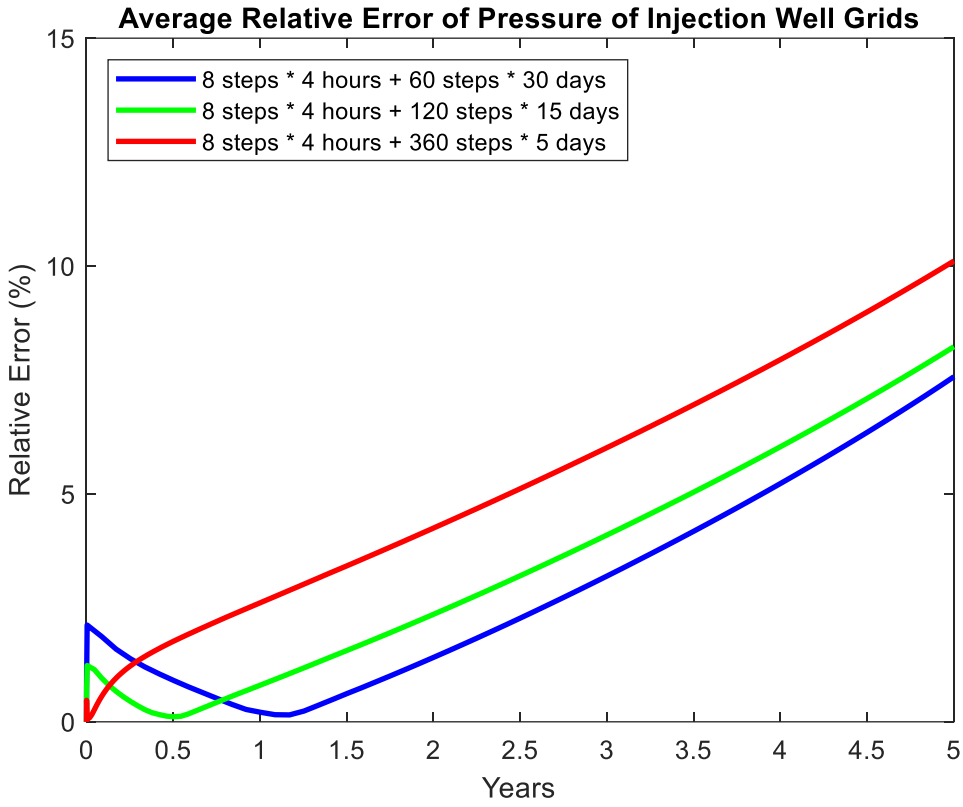


Figure 3.46 Average relative error of pressure for injection well grids in the geothermal model without validation.

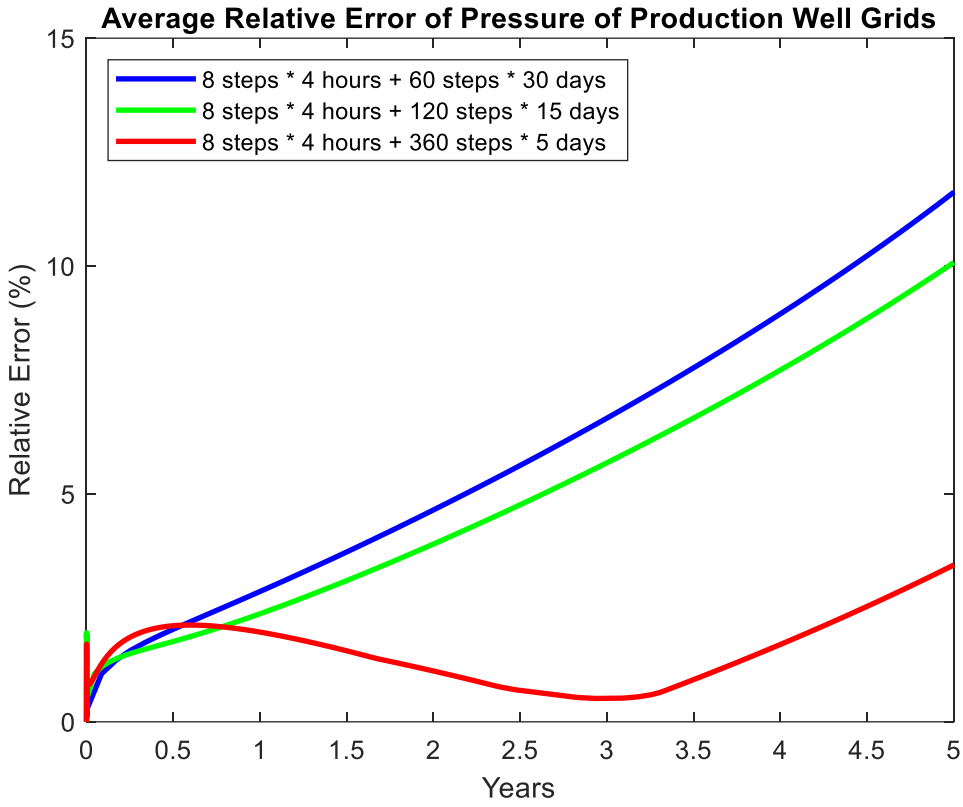


Figure 3.47 Average relative error of pressure for production well grids in the geothermal model without validation.

3.2.1.2. Temperature matching results on geothermal model without validation

The variation of temperature in the normal grids and production well grids are negligible, and the temperature in most of these grids almost holds constant (only a few grids near around the well has slightly temperature variation due to the injection of cold water). Therefore, the matching results and the corresponding average relative error for these grids are not shown.

Figure 3.48 shows the temperature matching results of the injection well grids for the three different timestep selections used. There is a sharp turning point on the temperature curve predicted by the NIROM method at the very beginning. This is because two different timestep intervals are

used in this model (4 hours timestep interval at the beginning and a longer timestep interval to the end). The prediction of temperature in the injection well grids by the proxy is clearly improved by using shorter timestep interval, and the sharp turning caused by the switching of timestep interval is also improved.

Figure 3.49 shows the average relative error of the prediction of temperature in the injection well grids. The case of 8 steps * 4 hours + 360 steps * 5 days has generated the highest accuracy, in which the average relative error is within 21.85%. The case with the worst accuracy is 8 steps * 4 hours + 60 steps * 30 days, and the highest average relative error can reach to 55.46%, which is not acceptable.

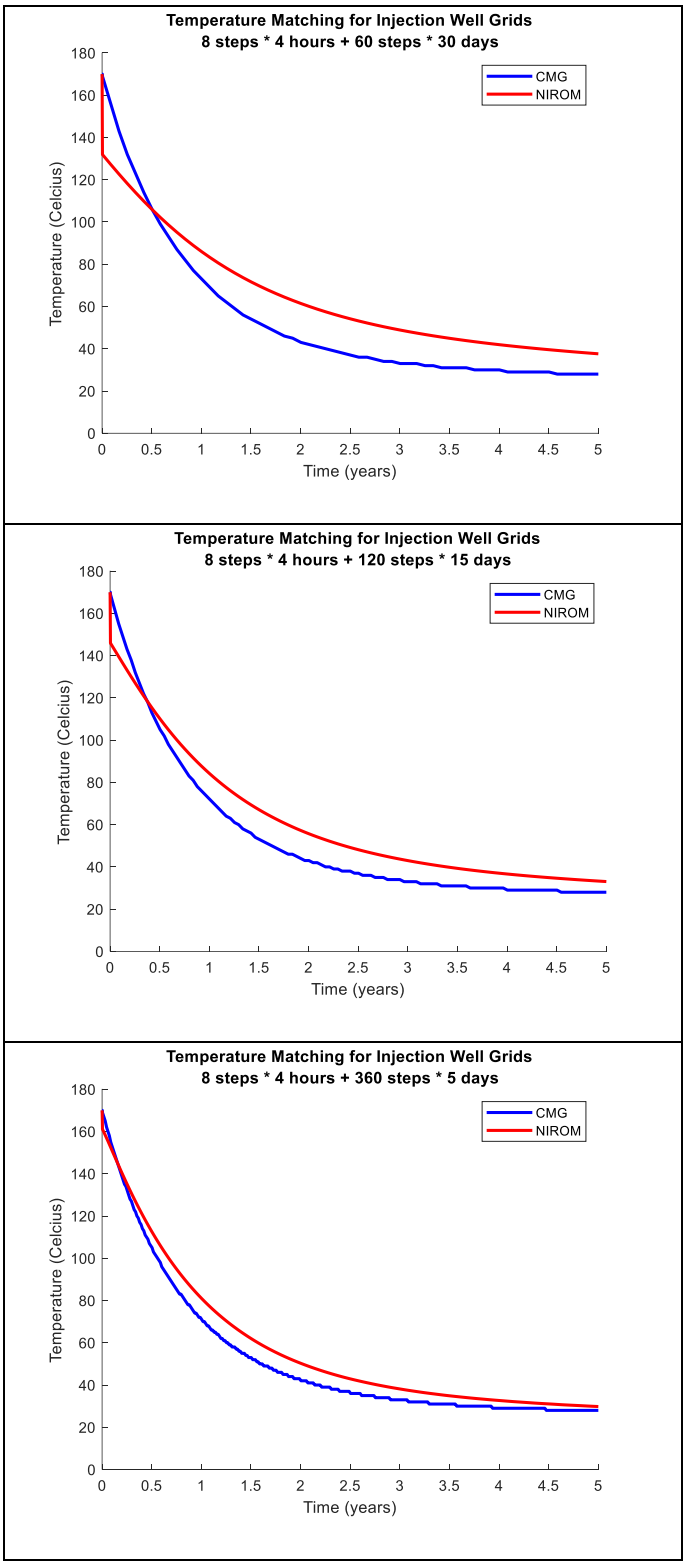


Figure 3.48 Temperature matching results for injection well grids of 3 different selections of timestep interval for the geothermal model without validation.

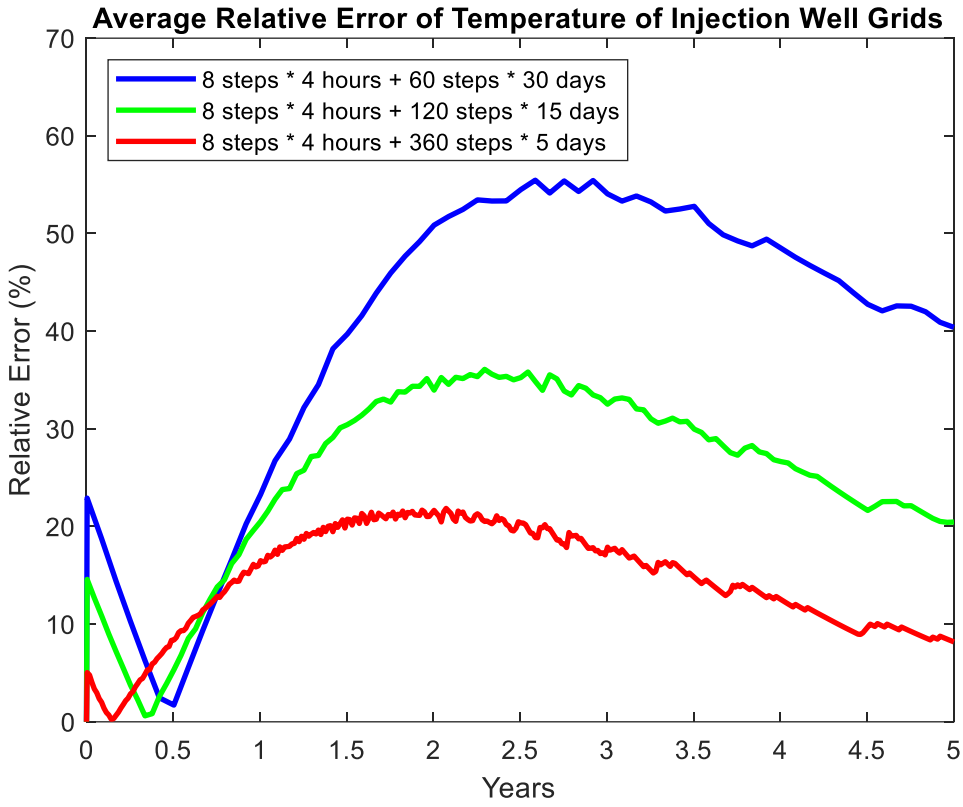


Figure 3.49 Average relative error of temperature for injection well grids in the geothermal model without validation.

3.2.1.3. Time reduction on geothermal model without validation

Table 3.8 presents the time data of implementing the NIROM method to the geothermal model without validation. All three cases have reduced the computational time significantly. The most accurate case (8 steps * 4 hours + 360 steps * 5 days) has time reduction of 92.56% with the average relative error within 10% in pressure prediction and within 21.85% in temperature prediction.

Table 3.8 Time data of the NIROM method for the geothermal model without validation.

Timesteps	High-Fidelity Commercial Simulator (CMG) Running Time (s)	Training Time (s)	Testing Time (s)	Computational Time Reduction Ratio
8 steps * 4 hours + 60 steps * 30 days	334	18.74	3.77	98.87%
8 steps * 4 hours + 120 steps * 15 days	584	59.95	12.88	97.79%
8 steps * 4 hours + 360 steps * 5 days	1425	457.05	105.99	92.56%

3.2.2. NIROM method on geothermal model with validation

The same as the procedure used in UNISIM-I-D model, a validation process is implemented following the algorithm introduced in the chapter of Methodology in the section of step 4 to help the NIROM method capture the multiple variation trends of the state. The flowrate of the wells used for building the validation set is shown in **Table 3.9**. Besides the additional validation set, the training set and testing set and the POD orders remain the same as before.

Table 3.9 The initial water flowrates for the training set, validation set, and the testing set for each well in the case study of the geothermal model with validation.

	Training Set Well Injection/Production Rate (m3/d)					Validation Set Well Injection/Production Rate (m3/d)	Testing Set Well Injection/Production Rate (m3/d)
Inj-1	625	675	725	775	825	700	650
Inj-2	625	675	725	775	825	700	650
Inj-3	625	675	725	775	825	700	650
Inj-4	625	675	725	775	825	700	650
Inj-5	625	675	725	775	825	700	650
Inj-6	625	675	725	775	825	700	650
Inj-7	625	675	725	775	825	700	650
Inj-8	625	675	725	775	825	2150	650
Prod-1	2000	2100	2200	2300	2400	2150	2350
Prod-2	2000	2100	2200	2300	2400	2150	2350
Prod-3	2000	2100	2200	2300	2400	2150	2350
Prod-4	2000	2100	2200	2300	2400	2150	2350
Prod-5	2000	2100	2200	2300	2400	2150	2350
Prod-6	2000	2100	2200	2300	2400	2150	2350
Prod-7	2000	2100	2200	2300	2400	2150	2350

3.2.2.1. Pressure matching results on geothermal model with validation

Figure 3.50 and **Figure 3.51** present the pressure matching results for the normal grids and the average relative error for the three different timestep selections when the validation process is implemented. The validation algorithm does not have obvious improvement on the matching results of the pressure curves obtained from the NIROM method and the high-fidelity model, because the pressure variation in the normal grids follows a smooth trend, and there is no sharp turning existing. The highest average relative error of pressure in normal grids remains below 9% in all three timestep setups.

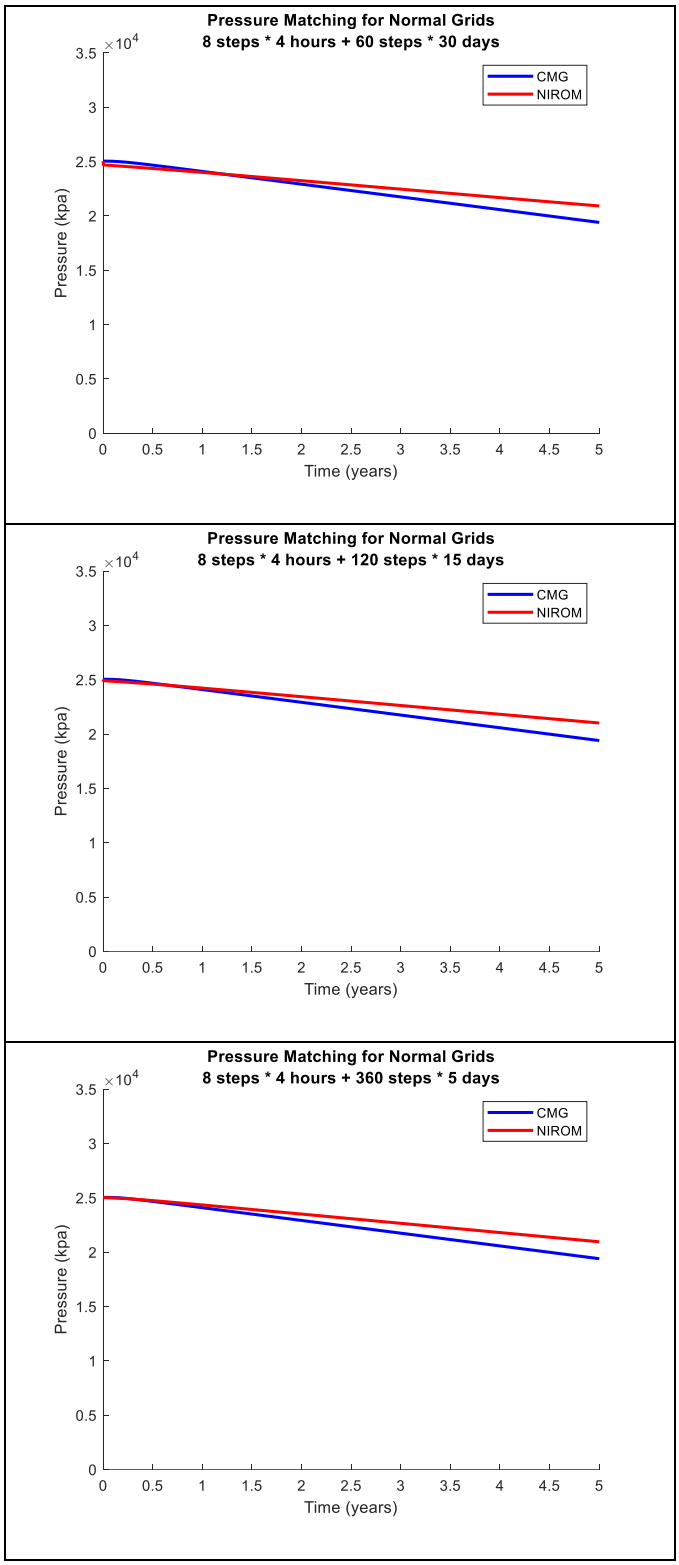


Figure 3.50 Pressure matching results for normal grids of 3 different selections of timestep interval for the geothermal model with validation.

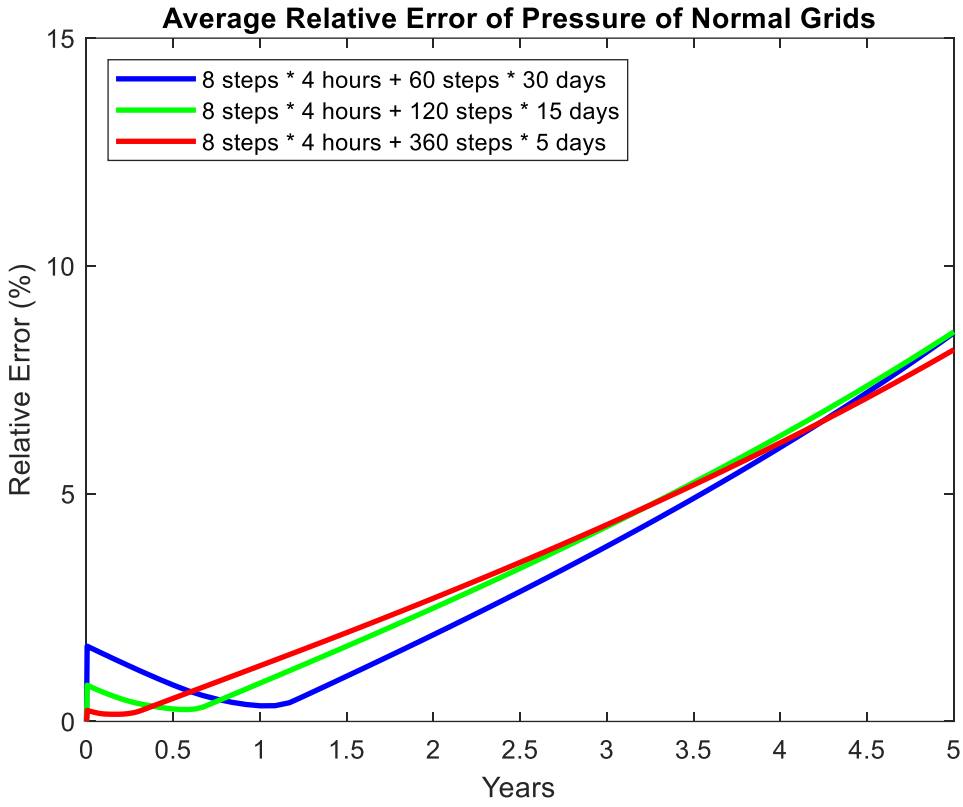


Figure 3.51 Average relative error of pressure for normal grids in the geothermal model with validation.

Figure 3.52 shows the pressure matching results in the injection well and production well grids for the three different timestep intervals, and **Figure 3.53** and **Figure 3.54** underneath show the average relative error of pressure of injection well and production well grids.

In the injection well grids, the average relative error is slightly improved. The average relative error in the case using 8 steps * 4 hours + 360 steps * 5 days decreases from 10.12% to 7.65%. At the same time, the average relative error of the production well grids does not have noticeable improvement. This is expected because the only turning in the variation trend of pressure is in the pressure of injection well grids. The pressure increases at first shortly due to injection and then

decreases because of draining from the reservoir. The algorithm successfully identifies this turning, and the accuracy in the injection well grids is slightly improved.

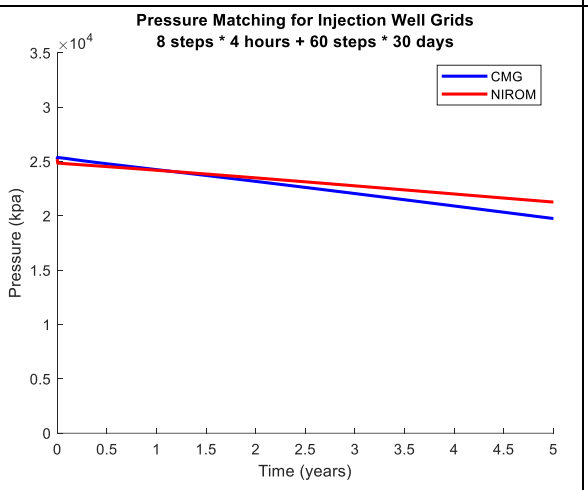
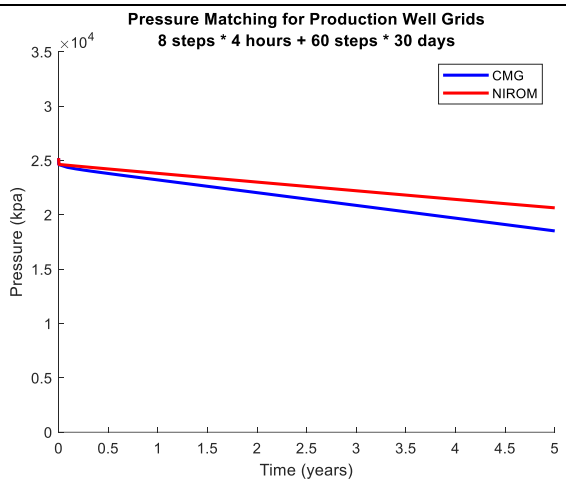
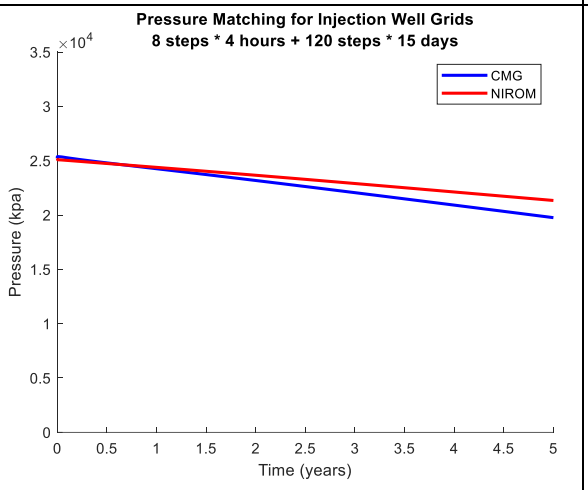
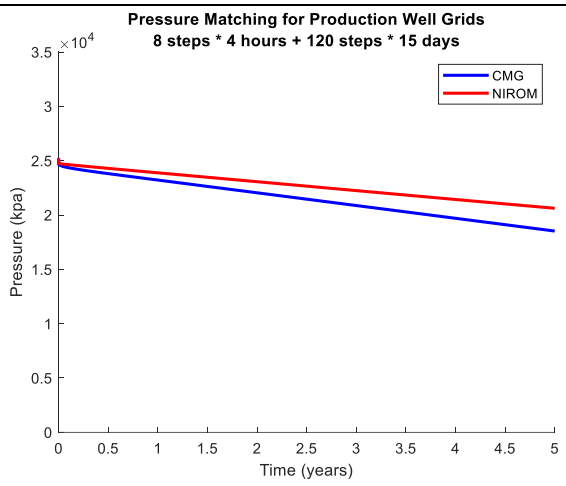
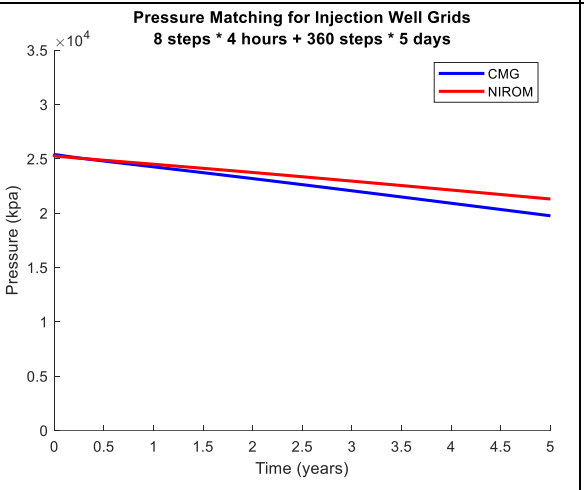
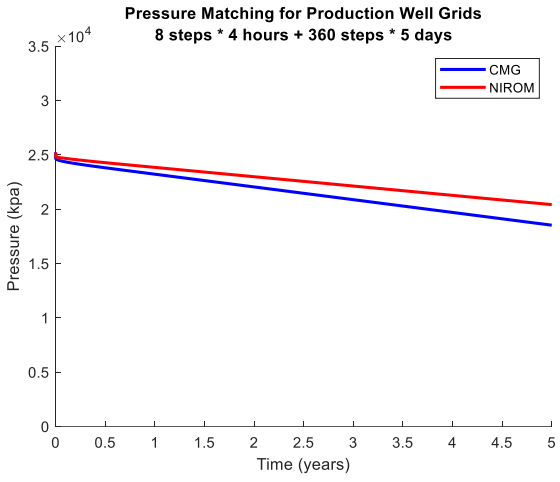
Pressure Matching Results for Injection Well Grids	Pressure Matching Results for Production Well Grids
	
	
	

Figure 3.52 Pressure matching results for well grids of 3 different selections of timestep interval for the geothermal model with validation.

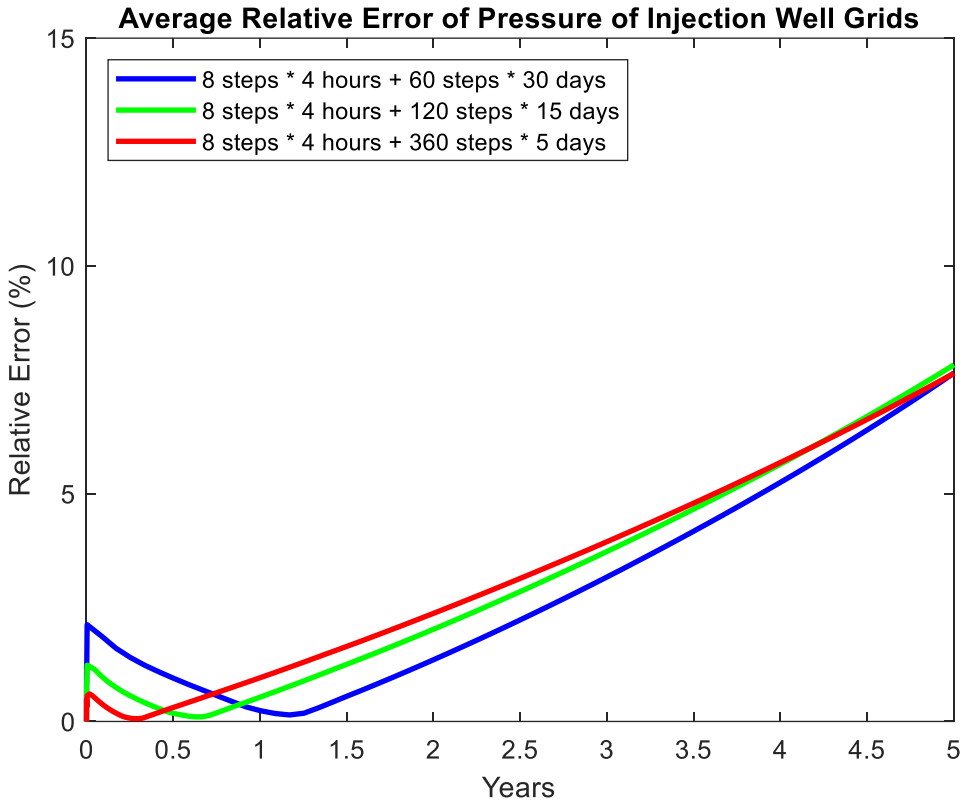


Figure 3.53 Average relative error of pressure for injection well grids in the geothermal model with validation.

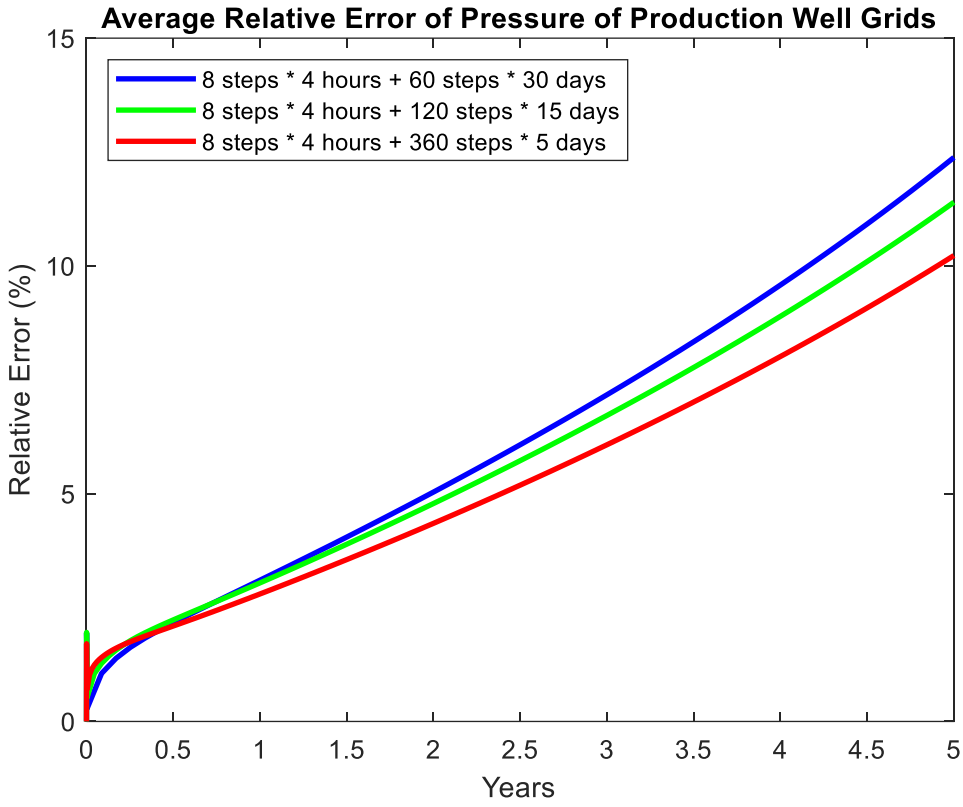


Figure 3.54 Average relative error of pressure for production well grids in the geothermal model with validation.

3.2.2.2. Temperature matching results on geothermal model with validation

Figure 3.55 presents the temperature matching results in the injection well grids, and **Figure 3.56** shows the average relative error. The matching results are slightly improved in all three cases. The highest average relative error decreases from 55.46% to 52.83% in the case 8 steps * 4 hours + 60 steps * 30 days, from 36.09% to 32.87% in the case 8 steps * 4 hours + 120 steps * 15 days, and from 21.85% to 17.26% in the case 8 steps * 4 hours + 360 steps * 5 days.

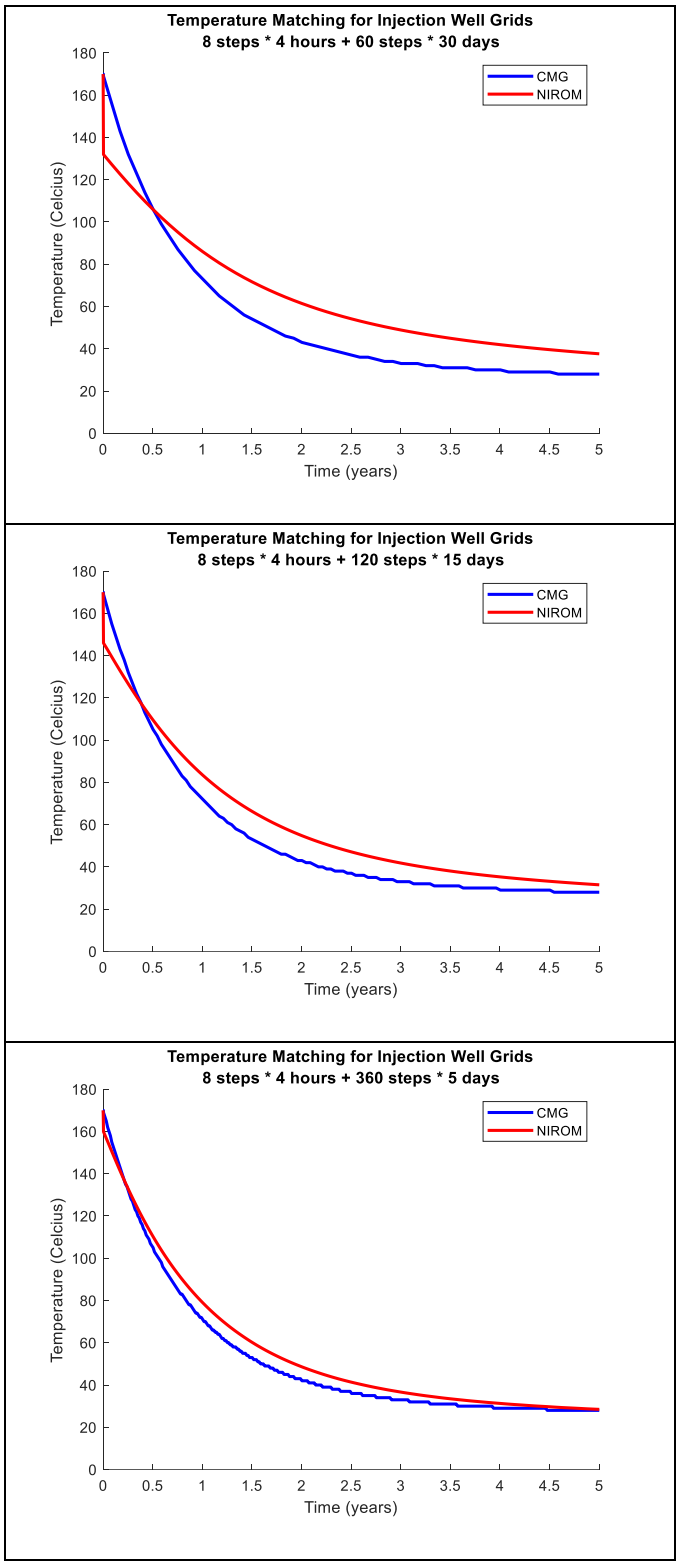


Figure 3.55 Temperature matching results for injection well grids of 3 different selections of timestep interval for the geothermal model with validation.

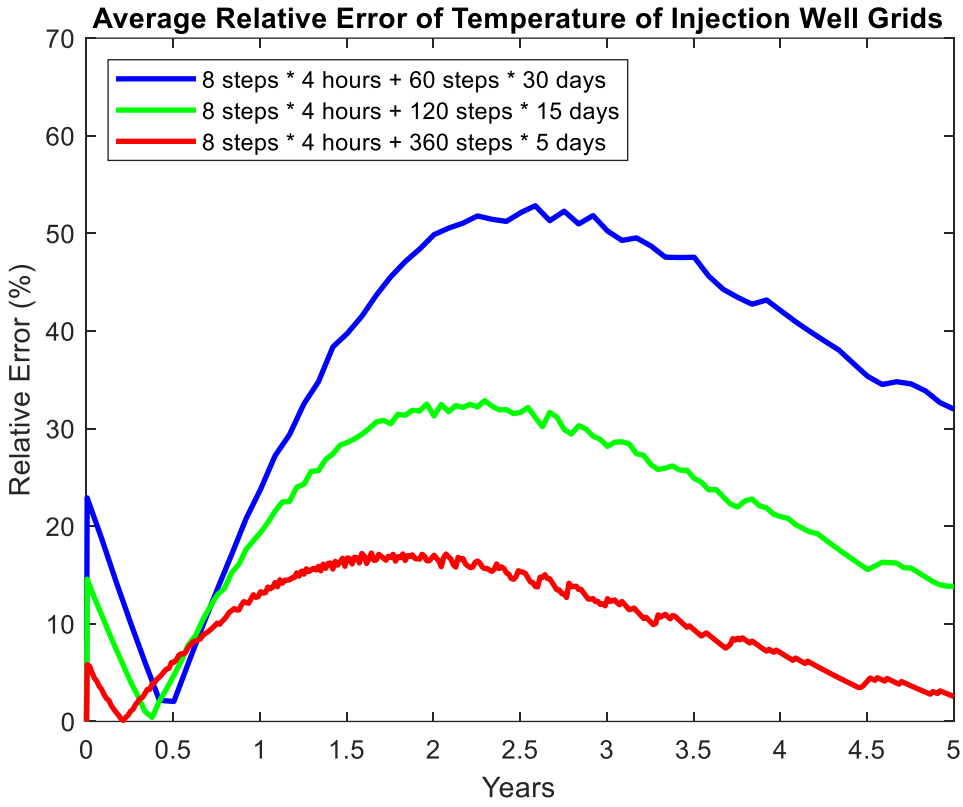


Figure 3.56 Average relative error of temperature for injection well grids in the geothermal model with validation.

3.2.2.3. Time reduction on geothermal model with validation

Table 3.10 shows the time data of implementing the NIROM method with the validation process to the geothermal model. Compared with **Table 3.8**, the offline computational time significantly increases for all the three cases. The computational time reduced ratio for each round of running the simulation with different setups does not vary too much, which coincide with the results of time reduction obtained in the UNISIM-I-D model.

Table 3.10 Time data of the NIROM method for the geothermal model with validation.

	High-Fidelity Commercial Simulator (CMG) Running Time (s)	Offline		Online	Computational Time Reduction Ratio
		Training Time (s)	Validation Time (s)	Testing Time (s)	
8 * 4 hours + 60 * 30 days	334	33.76	3.88	3.36	98.99%
8 * 4 hours + 120 * 15 days	584	112.23	12.85	12.63	97.84%
8 * 4 hours + 360 * 5 days	1425	889.69	102.8	111.67	92.16%

4. LIMITATIONS

When conducting reservoir simulation with the NIROM method using the surrogate model, the control imposed on the reservoir is input to the model in the format of control matrix. Noting that the surrogate model based on RBF is simply a combination of weighting factors and training data points if we describe the nature of it in one sentence, and therefore, it is not possible for it to generate reasonable prediction results for all the extreme conditions imposed. Thus, some requirements must be satisfied by the control matrix input. So far, the mathematical description of the boundary of whether the surrogate model can generate accurate prediction results on an arbitrary control matrix has not been thoroughly developed. However, the rule of thumb on the boundary of the control matrix that we can input into the surrogate model is that the value in the control matrix that we are predicting cannot beyond the range restricted by the minimum value and the maximum value in the control matrix used for training. For example, if we want to predict the variation of state variables triggered by a well with the oil flowrate of 1000 m^3/d, the training control matrix must contain flowrate values greater than 1000 m^3/d and less than 1000 m^3/d. The schematic illustration of the reason is shown in **Figure 4.1**. The solid curve is the surrogate model trained by RBF using the training data points. Remember that in RBF algorithm, a weighting factor correlated to each training point is computed basing on the distance and it represents the influence of the training data point to the value of the surrogate model (prediction). If a new data point (control matrix value) is close to the training data points, the distance from it to the training data points are small and we can thus correctly compute the influence of the training data points to the new data point. In contrast, if the new data point is far from the training data points, when we plug in the distance to the radial basis function, the value of it decrease very fast along with the increase in distance, which means the accuracy of the influence computed cannot be guaranteed. When

computing the value of a data point which is far from the training points, it is similar to computing the value of it using the extension of the trend of the surrogate model (grey dashed line in **Figure 4.1**) which may not be good enough to capture the nature of the system because of lack of training data points near around.

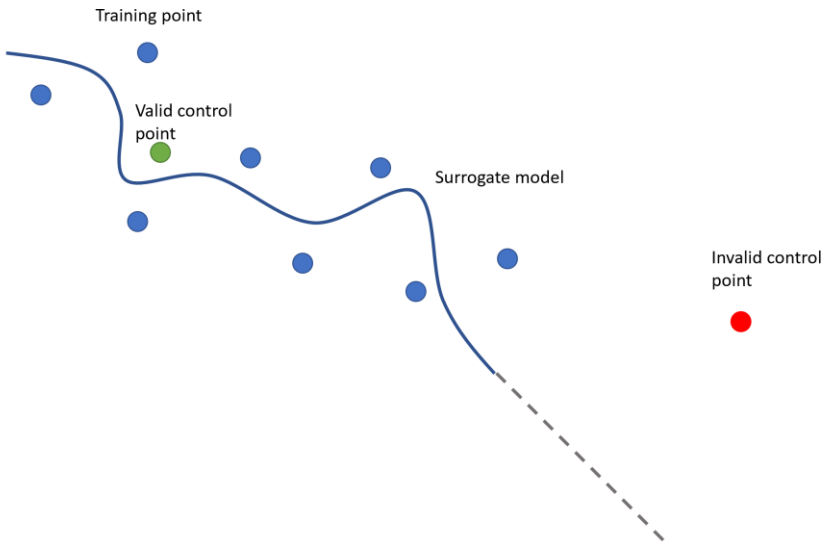


Figure 4.1 Schematic illustration of the requirement of the control matrix imposed to the surrogate model

Another factor that may limit the accuracy of the surrogate model is the number of training data set. In general, more training data sets used in training will result in more accurate prediction on the state variables, because the surrogate model is based on more knowledge of different controls. However, arbitrarily increasing the number of training data set will significantly affect the computational efficiency, because the weighting factor assigned to each training data point is computed by solving a linear equation, and the time expense of solving it increases exponentially with regards to the dimensions of it. Therefore, it is possible that in some very complex reservoir system, the surrogate model cannot obtain accurate prediction result even the computational time

of the surrogate model is greater than the high-fidelity simulator because of extremely large amount of training data points used.

5. CONCLUSIONS

5.1. Summary

In this work, a new non-intrusive reduced order modelling method is introduced. It has 5 steps which are snapshots collection, implementation of POD to obtain the training data set, implementation of RBF algorithm on the training set to obtain the proxy of the high-fidelity reservoir simulation model, the optional step of improving the performance of the proxy when multiple variation trends of the state exist with a validation process, testing the proxy on the testing data set built by setting up the system with different set of controls. The first 4 steps are conducted offline while the last step is online.

The NIROM method is implemented on two different case studies to evaluate both the accuracy and the ability of reducing computational time of the proxy. The first case study uses UNISIM-I-D reservoir model which is a 3-phase heterogeneous, isotropic, isothermal model. There are 4 production wells in this model which are producing at constant oil production rate at first, and then switch to constant bottom-hole pressure production. In this case study, the accuracy of the proxy is evaluated based on the prediction of pressure and oil saturation in both normal grids (no well passing through) and well grids. The second case study is on a geothermal 3-dimentional model which is homogenous and isotropic. In this case study, cold water is injected with a constant flowrate to the reservoir, and hot water is extracted from the reservoir at a constant flowrate as well. The accuracy of the proxy in this case study is evaluated on pressure in normal grids, injection well grids, and production well grids, as well as temperature in injection well grids. For each case study, three different timestep intervals used for collecting the snapshots of the states are compared.

In addition, the results of including the validation process and not including the validation process are compared.

5.2. Conclusions

The NIROM method has achieved the goal of predicting the states of the reservoir simulation model within an acceptable error in reduced computational time compared to the high-fidelity reservoir simulation model. More specifically, in the case study of UNISIM-I-D model, the proxy can reduce the computational time by 64.02% for each round of running. This time reduction causes at most 6.25% and 7.46% average relative error of pressure in the normal grids and well grids, 4.13% and 2.3% average relative error of oil saturation in normal grids and well grids, and 1.43%, 2.64%, 3.91%, 5.35% of relative error in cumulative oil production volume in well NA1A, NA2, NA3D, RJS19 respectively.

In the case study of the geothermal model, the proxy can reduce the computational time by 92.16% and causes at most 8.17%, 7.65%, 10.23% of average relative error of pressure in normal grids, injection well grids and production well grids respectively, and causes at most 17.26% average relative error of temperature in injection well grids.

In the two case studies, the proxy trained by using the shortest timestep interval to collect the snapshots for training has the highest accuracy in predicting the states in general. However, shorter timestep interval may result in the decrease in computational time reduction ratio. Therefore, the balance between the accuracy of the proxy and the time reduction on the model needs to be considered.

When multiple trends appear in the variation of state, the validation algorithm has successfully identified the turning points of the switch of these trends, and thus improves the performance of the proxy by training a group of surrogate models for which each surrogate model corresponds with one variation trend. In the case study of UNISIM-I-D model, the validation algorithm has improved the accuracy of the prediction of oil saturation. It has reduced the highest average relative

error of oil saturation from 6% to 4.13% in normal grids, and from 5.45% to 2.3% in production well grids. In the case study of the geothermal model, the validation algorithm has improved the performance of predicting pressure and temperature in injection well grids. The highest average relative error of pressure is reduced from 10.11% to 7.65%, and the highest average relative error of temperature is reduced from 21.85% to 17.26%.

5.3. Recommendations for future work

In this work, we have only discussed the feasibility of the NIROM method when the flowrate of the well needs to be adjusted. The other controls of the wells are kept the same when training the proxy with different flowrates. For example, the bottom-hole pressure of the well is also possible to be changed manually when solving optimization problems, which is not discussed in this work.

Also, in this work, the locations of the wells are fixed, and only vertical wells are considered in the case studies. However, in optimization problems, it is common to adjust the well locations and the inclination angle of the wells. The possibility of implementing this NIROM method to such cases needs to be explored in the future.

REFERENCES

1 Van Doren, J. F., Markovinović, R., & Jansen, J. D. (2006). Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Computational Geosciences*, *10*(1), 137-158.

2 Voloskov, D., & Pissarenko, D. (2020). Adaptive POD Galerkin technique for reservoir simulation and optimization. *arXiv preprint arXiv:2008.03471*.

3 Chaturantabut, S., & Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, *32*(5), 2737-2764.

4 Tan, X., Gildin, E., Florez, H., Trehan, S., Yang, Y., & Hoda, N. (2019). Trajectory-based DEIM (TDEIM) model reduction applied to reservoir simulation. *Computational Geosciences*, *23*(1), 35-53.

5 Cardoso, M. A., & Durlofsky, L. J. (2010). Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, *229*(3), 681-700.

6 Anifowose, F. A. (2011, January). Artificial intelligence application in reservoir characterization and modeling: whitening the black Box. In SPE Saudi Arabia section Young Professionals Technical Symposium. Society of Petroleum Engineers.

7 Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, *656*, 5-28.

8 Buhmann, M. D. (2000). Radial basis functions. *Acta numerica*, *9*, 1-38.

9 Schaback, R., & Wendland, H. (2006). Kernel techniques: from machine learning to meshless methods. *Acta numerica*, *15*, 543.

10 Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive approximation*, *2*(1), 11-22.

11 Xiao, D., Fang, F., Pain, C., & Hu, G. (2015). Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation. *International Journal for Numerical Methods in Fluids*, *79*(11), 580-595.

12 Fath, A. H., Madanifar, F., & Abbasi, M. (2020). Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems. *Petroleum*, *6*(1), 80-91.

13 Gaspar, A. T. F. S., Avansi, G. D., Santos, A. D. S., Hohendorff Filho, J. C. V., & Schiozer, D. J. (2015). UNISIM-ID: benchmark studies for oil field development and production strategy selection. Int. J. Model. Simul. Pet. Ind., 9(1), 47-55.

14 Ertekin, T., Abou-Kassem, J. H., & King, G. R. (2001). Basic applied reservoir simulation (Vol. 7). Richardson, TX: Society of Petroleum Engineers.

15 Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), 559-572.

16 Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. Psychometrika, 1(3), 211-218.

17 Zubarev, D. I. (2009, January). Pros and cons of applying proxy-models as a substitute for full reservoir simulations. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

18 Denney, D. (2010). Pros and cons of applying a proxy model as a substitute for full reservoir simulations. Journal of Petroleum Technology, 62(07), 41-42.

19 Jaber, A. K., Awang, M. B., & Lenn, C. P. (2017). Box-Behnken design for assessment proxy model of miscible CO2-WAG in heterogeneous clastic reservoir. Journal of Natural Gas Science and Engineering, 40, 236-248.

20 Chaki, S., Zagayevskiy, Y., Shi, X., Wong, T., & Noor, Z. (2020, January). Machine Learning for Proxy Modeling of Dynamic Reservoir Systems: Deep Neural Network DNN and Recurrent Neural Network RNN Applications. In International Petroleum Technology Conference. OnePetro.

21 Navrátil, J., King, A., Rios, J., Kollias, G., Torrado, R., & Codas, A. (2019). Accelerating physics-based simulations using end-to-end neural network proxies: An application in oil reservoir modeling. Frontiers in Big Data, 2, 33.