Hindawi
Geofluids
Volume 2018, Article ID 8482352, 15 pages
https://doi.org/10.1155/2018/8482352

Research Article

Acceleration of Gas Reservoir Simulation Using Proper Orthogonal Decomposition

Yi Wang,[1,2] Bo Yu,[3] and Ye Wang[4]

[1] *National Engineering Laboratory for Pipeline Safety and MOE Key Laboratory of Petroleum Engineering and Beijing Key Laboratory of Urban Oil and Gas Distribution Technology, China University of Petroleum, Beijing 102249, China*
[2] *Key Laboratory of Thermo-Fluid Science and Engineering, Xi'an Jiaotong University, Ministry of Education, Xi'an 710049, China*
[3] *School of Mechanical Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China*
[4] *Key Laboratory of Railway Vehicle Thermal Engineering, Lanzhou Jiaotong University, Ministry of Education, Lanzhou 730070, China*

Correspondence should be addressed to Bo Yu; yubobox@vip.163.com

Received 3 July 2017; Revised 26 August 2017; Accepted 17 September 2017; Published 3 January 2018

Academic Editor: Jianchao Cai

High-precision and high-speed reservoir simulation is important in engineering. Proper orthogonal decomposition (POD) is introduced to accelerate the reservoir simulation of gas flow in single-continuum porous media via establishing a reduced-order model by POD combined with Galerkin projection. Determination of the optimal mode number in the reduced-order model is discussed to ensure high-precision reconstruction with large acceleration. The typical POD model can achieve high precision for both ideal gas and real gas using only 10 POD modes. However, acceleration of computation can only be achieved for ideal gas. The obstacle of POD acceleration for real gas is that the computational time is mainly occupied by the equation of state (EOS). An approximation method is proposed to largely promote the computational speed of the POD model for real gas flow without decreasing the precision. The improved POD model shows much higher acceleration of computation with high precision for different reservoirs and different pressures. It is confirmed that the acceleration of the real gas reservoir simulation should use the approximation method instead of the computation of EOS.

1. Introduction

Reservoir simulation, utilizing mathematical models to predict fluid flow in petroleum reservoirs, has been developed since the 1800s [1]. It plays more and more important roles in the development of oil exploration and production [2]. Correspondingly, mathematical theories and numerical methods have attracted extensive attention [3–10]. The computational cost of the oil reservoir simulations is huge because a very dense mesh should be used causing long-time iterations to achieve enough resolution. This kind of long-time computation is usually not endurable in engineering. Petroleum engineers attempted several approaches to speed up oil reservoir simulations, including the scale-up approach [11–15] and parallel computation [16]. Proper orthogonal decomposition (POD) can reduce computational loads efficiently via a reduced-order model. It has been applied to heated-crude-oil pipe flow [17, 18] and other fields [19–21]. Ghommem et al. [22, 23] and Efendiev et al. [24] discussed the POD and dynamic mode decomposition method for time-dependent incompressible single-phase flow in high-contrast heterogeneous porous media. They proved that the POD model can achieve good precision in an appropriate range. With fast progress of the exploration and development of gas reservoirs, gas reservoir simulation attracted more and more attention [25–28]. Additional computation should be made to calculate the compressibility of the gas, which does not exist in incompressible oil flow. Thus, the numerical simulations of gas reservoirs are more complex and time-consuming. The study on the acceleration method of gas reservoir simulation by POD is meaningful. On the other hand, the compressibility of the gas leads to a POD model with stronger nonlinearity than oil flow. This situation might cause the POD modeling process to be more difficult than the previous POD modeling for incompressible single-phase flow. To the best of the authors' knowledge, there is no report concerning the POD model for gas reservoir simulation especially for real gas. Therefore, it is worth discussing the POD modeling method for gas reservoir simulations.

This paper is organized as follows. We revisit the governing equations of gas reservoir simulation and explain how to apply the POD to the gas reservoir simulation briefly. Then, a POD model will be established for gas reservoir simulation in single-continuum porous media. Key effects will be discussed on the acceleration ability of the POD model. Finally, conclusions and suggestions will be made.

2. Numerical Methods

2.1. Governing Equations of Gas Reservoir Simulation. Pure gas flow obeys mass conservation law (see (1)), Darcy's law (see (2)), and the gas law (see (3)) as follows:

$$\phi\frac{\partial\rho}{\partial t} = -\nabla\cdot(\rho\mathbf{u}) + q, \tag{1}$$

$$\mathbf{u} = -\frac{\mathbf{k}}{\mu}\nabla p, \tag{2}$$

$$\rho = \frac{pW}{ZRT}, \tag{3}$$

where $\mathbf{u}$ is the Darcy velocity of gas flow in reservoirs, $\mathbf{k}$ is the diagonal permeability tensor, p is the pressure, ρ is the density of gas, μ is the dynamic viscosity of gas, ϕ is the porosity of porous media, q is the injection or production rate, W is the molecular weight of gas, R is the universal gas constant, T is the temperature of gas, and Z is the compressibility factor of gas. Z should be calculated by the equation of state of gas. In this paper, we use the Van der Waals equation:

$$Z^3 - (B+1)Z^2 + AZ - AB = 0, \tag{4}$$

where $A = ap/R^2T^2$, $B = bp/RT$, $Z = pv/RT$, v is the molar volume of gas, and a and b are gas constants changing with different real gases. The compressibility of gas for isothermal flow can be defined as

$$c_g = \frac{1}{\rho}\left.\frac{d\rho}{dp}\right|_T. \tag{5}$$

Using pressure as the primary variable, (1) and (5) can be rewritten as follows:

$$\phi c_g p\frac{\partial p}{\partial t} = \nabla\cdot\left(\frac{\mathbf{k}}{\mu}p\nabla p\right) + \frac{ZRT}{W}q, \tag{6}$$

$$c_g = \frac{1}{p} - \frac{1}{Z}\left[\frac{B-Z}{3Z^2-2(B+1)Z+A}\frac{a}{R^2T^2} + \frac{A+Z^2}{3Z^2-2(B+1)Z+A}\frac{b}{RT}\right]. \tag{7}$$

For the convenience of expression, we only use two-dimensional cases in this paper, but the extension to three-dimensional cases is straightforward. Thus, the above governing equations have the following expressions:

$$\phi c_g p\frac{\partial p}{\partial t} = \frac{\partial}{\partial x}\left(\frac{k_{xx}}{\mu}p\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{k_{yy}}{\mu}p\frac{\partial p}{\partial y}\right) + \frac{ZRT}{W}q, \tag{8}$$

$$u = -\frac{k_{xx}}{\mu}\frac{\partial p}{\partial x}, \tag{9}$$

$$v = -\frac{k_{yy}}{\mu}\frac{\partial p}{\partial y}, \tag{10}$$

where k_{xx} and k_{yy} are two components of permeability $\mathbf{k}$ in the x and y directions, respectively, and u and v are two components of Darcy velocity $\mathbf{u}$ in the x and y directions, respectively. Gas reservoir simulation can be accomplished via numerically solving (8)~(10) to obtain pressure and velocity with the calculation of parameters (Z and c_g) using (4) and (7). Here, c_g is the implicit function of p (shown in (7)) so that it can only be explicitly solved. Therefore, the whole term $c_g p$ is treated explicitly.

2.2. Basic Idea of Proper Orthogonal Decomposition. The above governing equations clearly show that the core of the whole computation is the calculation of p. Once p is solved, one can directly solve u and v. Therefore, the acceleration of the computation of p is the most important in the POD modeling. The basic idea of the POD modeling for (8) is briefly stated as follows.

(1) Separate the variable in space and time using the following linear combination:

$$p = \sum_{n=1}^{M} c_n\varphi_n, \tag{11}$$

where φ_n are the POD modes which are the functions of space ($\varphi_n(x,y)$), c_n are the temporal coefficients ($c_n(t)$), and φ_n are the identity basis vectors orthogonal with each other:

$$\varphi_i\cdot\varphi_j = \delta_{ij}, \tag{12}$$

where "$\cdot$" means the inner product of two vectors and δ_{ij} is the Kronecker delta taking the value 1 for $j = i$ and 0 for others.

(2) φ_n can be calculated from the eigenvalue decomposition of a sample matrix, which can be obtained by aligning the samples in a time series $(t_1, t_2, t_3, \ldots, t_N)$:

$$\mathbf{S} = [S_1, S_2, S_3, \ldots, S_N], \tag{13}$$

where $S_1 = \begin{bmatrix} p_{11} \\ p_{21} \\ \vdots \\ p_{L1}\end{bmatrix}$, $S_2 = \begin{bmatrix} p_{12} \\ p_{22} \\ \vdots \\ p_{L2}\end{bmatrix}$, $S_3 = \begin{bmatrix} p_{13} \\ p_{23} \\ \vdots \\ p_{L3}\end{bmatrix}, \ldots, S_N = \begin{bmatrix} p_{1N} \\ p_{2N} \\ \vdots \\ p_{LN}\end{bmatrix}$, L is the number of grid points (including boundary points), and N is the number of samples. A kernel can be calculated as follows:

$$C_{ij} = \frac{1}{N}\int_\Omega S_iS_j\,d\Omega, \tag{14}$$

where Ω means domain. The matrix form of (14) is

$$\mathbf{C} = \frac{\Delta x \Delta y}{N} \mathbf{S}^T \mathbf{S}. \tag{15}$$

The matrix **C** can be decomposed by eigenvalue decomposition to obtain the eigenvalues and eigenvectors:

$$\mathbf{C}\mathbf{V} = \lambda \mathbf{V}, \tag{16}$$

where **V** is the eigenvector matrix and λ are the eigenvalues of the corresponding eigenvectors. Along with the descending order of λ ($\lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_N$), the importance of eigenvectors $v_1, v_2, v_3, \ldots, v_N$ in **V** is also descending. To quantitatively represent the importance of each eigenvector, the energy ξ_n of the nth eigenvector is introduced:

$$\xi_n = \frac{\lambda_n}{\sum_{i=1}^{N} \lambda_i} \times 100\%. \tag{17}$$

Correspondingly, the cumulative energy ζ_n of the top n eigenvectors can be expressed as follows:

$$\zeta_n = \frac{\sum_{j=1}^{n} \lambda_j}{\sum_{i=1}^{N} \lambda_i} \times 100\%. \tag{18}$$

POD modes φ_n can be obtained as follows:

$$\varphi_n = \frac{\mathbf{S}\mathbf{V}}{\|\mathbf{S}\mathbf{V}\|}, \tag{19}$$

where "‖‖" means the Euclidean distance.

(3) Substituting (11) into (8) and projecting the whole equation onto the Hilbert space spanned by the POD modes ϕ_m ($m = 1 \sim M$), one can finally obtain a linear equation system of c_n in the scale of $M \times M$, where M is the number of POD modes in (11). Once the temporal coefficients c_n are solved, the original variable p can be directly reconstructed in (11), instead of the computation of (8). The computational cost of (11) is in the scale of $M \times M$ while the computational cost of (8) is in the scale of $L \times L$. The number of samples is usually quite smaller than the number of grid points ($N \ll L$). The number of effective POD modes is smaller than the number of samples ($M \leq N$). Thus, the computational time can be largely reduced using POD. The POD model is a type of reduced-order model. The details of this step will be explained in Section 2.3.

According to the above analyses, the core computation of POD is the calculation of temporal coefficients c_n. The equation of c_n for gas reservoir simulation will be established in detail in the next section.

2.3. Establishment of the POD Model for Gas Reservoir Simulation.

Substituted by (11), (8) can be transformed to

$$\begin{aligned} &\phi c_g p \sum_{n=1}^{M} \varphi_n \frac{dc_n}{dt} \\ &\quad = \sum_{n=1}^{M} \sum_{l=1}^{M} c_n c_l \left[\frac{\partial}{\partial x} \left(\frac{k_{xx}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{k_{yy}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial y} \right) \right] \\ &\quad + \frac{ZRT}{W} q. \end{aligned} \tag{20}$$

Equation (20) is projected onto φ_m:

$$\int_0^{ly} \int_0^{lx} \varphi_m \phi c_g p \sum_{n=1}^{M} \varphi_n \frac{dc_n}{dt} dx\, dy = \int_0^{ly} \int_0^{lx} \varphi_m \sum_{n=1}^{M} \sum_{l=1}^{M} c_n c_l \left[\frac{\partial}{\partial x} \left(\frac{k_{xx}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{k_{yy}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial y} \right) \right] dx\, dy + \int_0^{ly} \int_0^{lx} \varphi_m \frac{ZRT}{W} q\, dx\, dy, \tag{21}$$

where lx and ly are the lengths of the computational domain in the x and y directions, respectively. As stated in (11), c_n and φ_n are functions of time and space, respectively. Thus, the left-hand side of (21) is transformed to

$$\int_0^{ly} \int_0^{lx} \varphi_m \phi c_g p \sum_{n=1}^{M} \varphi_n \frac{dc_n}{dt} dx\, dy = \phi \sum_{n=1}^{M} \frac{dc_n}{dt} \int_0^{ly} \int_0^{lx} c_g p \varphi_m \varphi_n dx\, dy. \tag{22}$$

Using integration by part and the property of the POD modes (see (12)), we can simplify the first term and the second term of the right-hand side of (21) as follows:

$$\begin{aligned} &\int_0^{ly} \int_0^{lx} \varphi_m \sum_{n=1}^{M} \sum_{l=1}^{M} c_n c_l \left[\frac{\partial}{\partial x} \left(\frac{k_{xx}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{k_{yy}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial y} \right) \right] dx\, dy \\ &\quad = \sum_{n=1}^{M} \sum_{l=1}^{M} c_n c_l \int_0^{ly} \int_0^{lx} \varphi_m \left[\frac{\partial}{\partial x} \left(\frac{k_{xx}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{k_{yy}}{\mu} \varphi_l \frac{\partial \varphi_n}{\partial y} \right) \right] dx\, dy = \int_0^{lx} \Bigg[\left(\varphi_m \frac{k_{yy}}{\mu} p \frac{\partial p}{\partial y} \right)_{ly} \end{aligned}$$

$$
\begin{aligned}
&-\left(\varphi_m \frac{k_{yy}}{\mu} p \frac{\partial p}{\partial y}\right)_0\Bigg] dx + \int_0^{ly} \left[\left(\varphi_m \frac{k_{xx}}{\mu} p \frac{\partial p}{\partial x}\right)_{lx} - \left(\varphi_m \frac{k_{xx}}{\mu} p \frac{\partial p}{\partial x}\right)_0\right] dy \\
&- \sum_{n=1}^{M}\sum_{l=1}^{M} c_n c_l \int_0^{ly}\int_0^{lx} \varphi_l \left(\frac{k_{xx}}{\mu}\frac{\partial \varphi_n}{\partial x}\frac{\partial \varphi_m}{\partial x} + \frac{k_{yy}}{\mu}\frac{\partial \varphi_n}{\partial y}\frac{\partial \varphi_m}{\partial y}\right) dx\, dy,
\end{aligned}
\tag{23}
$$

$$
\int_0^{ly}\int_0^{lx} \varphi_m \frac{ZRT}{W} q\, dx\, dy = \frac{RT}{W}\int_0^{ly}\int_0^{lx} \varphi_m Z q\, dx\, dy. \tag{24}
$$

Please note that the projection of the boundary condition in (23) is dependent on boundary pressures and boundary pressure gradients. To update them, the pressure field should be firstly reconstructed using (11) in every time step once c_n is updated by the POD model. Therefore, the projection of the boundary condition leads to additional computations of the full-order equations (see (9), (10), and (11)), consuming much more time. This is very harmful for the high acceleration of the POD model. To ensure the high-acceleration advantage of the POD model, the computations of (9), (10), and (11) should be avoided in the POD model. This can be fulfilled by further decomposing the pressures in (23). The derivations will be made in the following three steps.

(1) If all the boundaries are Dirichlet condition (known boundary pressure), boundary treatment can be expressed as follows:

$$
\begin{aligned}
\left(\frac{\partial p}{\partial x}\right)_{nx+1,j} &= \frac{p_{nx+1,j} - p_{nx,j}}{\Delta x/2} \\
&= \frac{2}{\Delta x}\left(p_{nx+1,j} - \sum_{n=1}^{M} c_n (\varphi_n)_{nx,j}\right), \\
\left(\frac{\partial p}{\partial x}\right)_{0,j} &= \frac{p_{1,j} - p_{0,j}}{\Delta x/2} \\
&= \frac{2}{\Delta x}\left(\sum_{n=1}^{M} c_n (\varphi_n)_{1,j} - p_{0,j}\right), \\
\left(\frac{\partial p}{\partial x}\right)_{i,ny+1} &= \frac{p_{i,ny+1} - p_{i,ny}}{\Delta y/2} \\
&= \frac{2}{\Delta y}\left(p_{i,ny+1} - \sum_{n=1}^{M} c_n (\varphi_n)_{i,ny}\right), \\
\left(\frac{\partial p}{\partial x}\right)_{i,0} &= \frac{p_{i,1} - p_{i,0}}{\Delta y/2} \\
&= \frac{2}{\Delta y}\left(\sum_{n=1}^{M} c_n (\varphi_n)_{i,1} - p_{i,0}\right),
\end{aligned}
\tag{25}
$$

where the boundary pressures $p_{nx+1,j}$, $p_{0,j}$, $p_{i,ny+1}$, and $p_{i,0}$ are constant so that they remain in the expressions, and their neighbor pressures ($p_{nx,j}$, $p_{1,j}$, $p_{i,ny}$, $p_{i,1}$) are changing with time so that they are decomposed into c_n and φ_n using (11). Thus, the projection of the boundary condition can be transformed to

$$
\begin{aligned}
&\int_0^{lx}\left[\left(\varphi_m \frac{k_{yy}}{\mu} p \frac{\partial p}{\partial y}\right)_{ly} - \left(\varphi_m \frac{k_{yy}}{\mu} p \frac{\partial p}{\partial y}\right)_0\right] dx \\
&+ \int_0^{ly}\left[\left(\varphi_m \frac{k_{xx}}{\mu} p \frac{\partial p}{\partial x}\right)_{lx} - \left(\varphi_m \frac{k_{xx}}{\mu} p \frac{\partial p}{\partial x}\right)_0\right] dy \\
&= \frac{2\Delta y}{\Delta x}\sum_{j=1}^{ny}\left[\left(\frac{k_{xx}}{\mu}\varphi_m p^2\right)_{nx+1,j} + \left(\frac{k_{xx}}{\mu}\varphi_m p^2\right)_{0,j}\right] \\
&+ \frac{2\Delta x}{\Delta y}\sum_{i=1}^{nx}\left[\left(\frac{k_{yy}}{\mu}\varphi_m p^2\right)_{i,ny+1} + \left(\frac{k_{yy}}{\mu}\varphi_m p^2\right)_{i,0}\right] \\
&- \sum_{n=1}^{M} c_n \left\{\frac{2\Delta y}{\Delta x}\sum_{j=1}^{ny}\left[\left(\frac{k_{xx}}{\mu}\varphi_m p\right)_{nx+1,j} (\varphi_n)_{nx,j}\right.\right. \\
&\left. + \left(\frac{k_{xx}}{\mu}\varphi_m p\right)_{0,j} (\varphi_n)_{1,j}\right] + \frac{2\Delta x}{\Delta y} \\
&\cdot \sum_{i=1}^{nx}\left[\left(\frac{k_{yy}}{\mu}\varphi_m p\right)_{i,ny+1} (\varphi_n)_{i,ny}\right. \\
&\left.\left. + \left(\frac{k_{yy}}{\mu}\varphi_m p\right)_{i,0} (\varphi_n)_{i,1}\right]\right\}.
\end{aligned}
\tag{26}
$$

(2) If all the boundaries are Neumann condition (known boundary velocity), boundary treatment can be expressed as follows:

$$
\begin{aligned}
u_{0,j} &= -\left(\frac{k_{xx}}{\mu}\frac{\partial p}{\partial x}\right)_{0,j}, \\
u_{nx,j} &= -\left(\frac{k_{xx}}{\mu}\frac{\partial p}{\partial x}\right)_{nx+1,j}, \\
v_{i,0} &= -\left(\frac{k_{yy}}{\mu}\frac{\partial p}{\partial y}\right)_{i,0}, \\
v_{i,ny} &= -\left(\frac{k_{yy}}{\mu}\frac{\partial p}{\partial y}\right)_{i,ny+1}.
\end{aligned}
\tag{27}
$$

The projection of the boundary condition can be transformed to

$$\begin{aligned}
&\int_0^{lx}\left[\left(\varphi_m\frac{k_{yy}}{\mu}p\frac{\partial p}{\partial y}\right)_{ly}-\left(\varphi_m\frac{k_{yy}}{\mu}p\frac{\partial p}{\partial y}\right)_0\right]dx \\
&\quad+\int_0^{ly}\left[\left(\varphi_m\frac{k_{xx}}{\mu}p\frac{\partial p}{\partial x}\right)_{lx}-\left(\varphi_m\frac{k_{xx}}{\mu}p\frac{\partial p}{\partial x}\right)_0\right]dy \\
&\quad=\sum_{n=1}^{M}c_n \\
&\quad\cdot\left\{\sum_{j=1}^{ny}\left[u_{0,j}\left(\varphi_m\varphi_n\right)_{0,j}-u_{nx,j}\left(\varphi_m\varphi_n\right)_{nx+1,j}\right]\Delta y\right. \\
&\quad\left.+\sum_{i=1}^{nx}\left[v_{i,0}\left(\varphi_m\varphi_n\right)_{i,0}-v_{i,ny}\left(\varphi_m\varphi_n\right)_{i,ny+1}\right]\Delta x\right\}.
\end{aligned} \tag{28}$$

(3) For general boundary conditions, the projection of the boundary condition can be summarized via both (26) and (28):

$$\begin{aligned}
&\int_0^{lx}\left[\left(\varphi_m\frac{k_{yy}}{\mu}p\frac{\partial p}{\partial y}\right)_{ly}-\left(\varphi_m\frac{k_{yy}}{\mu}p\frac{\partial p}{\partial y}\right)_0\right]dx \\
&\quad+\int_0^{ly}\left[\left(\varphi_m\frac{k_{xx}}{\mu}p\frac{\partial p}{\partial x}\right)_{lx}-\left(\varphi_m\frac{k_{xx}}{\mu}p\frac{\partial p}{\partial x}\right)_0\right]dy \\
&\quad=HB_m+\sum_{n=1}^{M}c_n\left(HV_{m,n}-HP_{m,n}\right),
\end{aligned} \tag{29}$$

where

$$\begin{aligned}
HB_m&=\frac{2\Delta y}{\Delta x}\sum_{j=1}^{ny}\left[\left(\frac{k_{xx}}{\mu}\varphi_m p^2\right)_{nx+1,j}\text{Diri}\,X_{nx+1,j}\right. \\
&\quad\left.+\left(\frac{k_{xx}}{\mu}\varphi_m p^2\right)_{0,j}\text{Diri}\,X_{0,j}\right]+\frac{2\Delta x}{\Delta y} \\
&\quad\cdot\sum_{i=1}^{nx}\left[\left(\frac{k_{yy}}{\mu}\varphi_m p^2\right)_{i,ny+1}\text{Diri}\,Y_{i,ny+1}\right. \\
&\quad\left.+\left(\frac{k_{yy}}{\mu}\varphi_m p^2\right)_{i,0}\text{Diri}\,Y_{i,0}\right], \\
HV_{m,n}&=\Delta y\sum_{j=1}^{ny}\left[u_{0,j}\left(\varphi_m\varphi_n\right)_{0,j}\left(1-\text{Diri}\,X_{0,j}\right)\right. \\
&\quad\left.-u_{nx,j}\left(\varphi_m\varphi_n\right)_{nx+1,j}\left(1-\text{Diri}\,X_{nx+1,j}\right)\right] \\
&\quad+\Delta x\sum_{i=1}^{nx}\left[v_{i,0}\left(\varphi_m\varphi_n\right)_{i,0}\left(1-\text{Diri}\,Y_{i,0}\right)\right. \\
&\quad\left.-v_{i,ny}\left(\varphi_m\varphi_n\right)_{i,ny+1}\left(1-\text{Diri}\,Y_{i,ny+1}\right)\right], \\
HP_{m,n}&=\frac{2\Delta y}{\Delta x} \\
&\quad\cdot\sum_{j=1}^{ny}\left[\left(\frac{k_{xx}}{\mu}\varphi_m p\right)_{nx+1,j}\left(\varphi_n\right)_{nx,j}\text{Diri}\,X_{nx+1,j}\right. \\
&\quad\left.+\left(\frac{k_{xx}}{\mu}\varphi_m p\right)_{0,j}\left(\varphi_n\right)_{1,j}\text{Diri}\,X_{0,j}\right]+\frac{2\Delta x}{\Delta y} \\
&\quad\cdot\sum_{i=1}^{nx}\left[\left(\frac{k_{yy}}{\mu}\varphi_m p\right)_{i,ny+1}\left(\varphi_n\right)_{i,ny}\text{Diri}\,Y_{i,ny+1}\right. \\
&\quad\left.+\left(\frac{k_{yy}}{\mu}\varphi_m p\right)_{i,0}\left(\varphi_n\right)_{i,1}\text{Diri}\,Y_{i,0}\right],
\end{aligned} \tag{30}$$

where Diri X and Diri Y are 1 for Dirichlet boundary condition and 0 for Neumann boundary condition. Thus, boundary pressures only appear in the expressions of HB_m and $HP_{m,n}$ while boundary velocities only appear in the expression of $HV_{m,n}$. HB_m, $HV_{m,n}$, and $HP_{m,n}$ can also be calculated only once, saving computational time. Let

$$\begin{aligned}
HU_{m,n}&=\int_0^{ly}\int_0^{lx}c_g p\varphi_m\varphi_n dx\,dy \\
&=\sum_{j=1}^{ny}\sum_{i=1}^{nx}\left(c_g p\varphi_m\varphi_n\right)_{i,j}\Delta x\Delta y,
\end{aligned} \tag{31}$$

$$\begin{aligned}
&HD_{m,n,l} \\
&\quad=\int_0^{ly}\int_0^{lx}\varphi_l\left(\frac{k_{xx}}{\mu}\frac{\partial\varphi_n}{\partial x}\frac{\partial\varphi_m}{\partial x}+\frac{k_{yy}}{\mu}\frac{\partial\varphi_n}{\partial y}\frac{\partial\varphi_m}{\partial y}\right)dx\,dy \\
&\quad=\sum_{j=1}^{ny}\sum_{i=1}^{nx}\left[\varphi_l\left(\frac{k_{xx}}{\mu}\frac{\partial\varphi_n}{\partial x}\frac{\partial\varphi_m}{\partial x}+\frac{k_{yy}}{\mu}\frac{\partial\varphi_n}{\partial y}\frac{\partial\varphi_m}{\partial y}\right)\right]_{i,j}\Delta x\Delta y,
\end{aligned} \tag{32}$$

$$HS_m=\int_0^{ly}\int_0^{lx}\varphi_m Zq\,dx\,dy=\sum_{j=1}^{ny}\sum_{i=1}^{nx}\left(\varphi_m Zq\right)_{i,j}\Delta x\Delta y. \tag{33}$$

Then, the POD model for gas reservoir simulation can be obtained in the following expression:

$$\begin{aligned}
\phi\sum_{n=1}^{M}\frac{dc_n}{dt}HU_{m,n}&=HB_m+\sum_{n=1}^{M}c_n\left(HV_{m,n}-HP_{m,n}\right) \\
&\quad-\sum_{n=1}^{M}\sum_{l=1}^{M}c_n c_l HD_{m,n,l}+\frac{RT}{W}HS_m.
\end{aligned} \tag{34}$$

Equation (34) is the time evolution equation of c_n. For each time step, (34) can be solved via an efficient linear solver (DLSARG) provided by FORTRAN coding language. Numerical results are discussed in the next section to evaluate the precision and the acceleration performances of the POD reduced-order model.

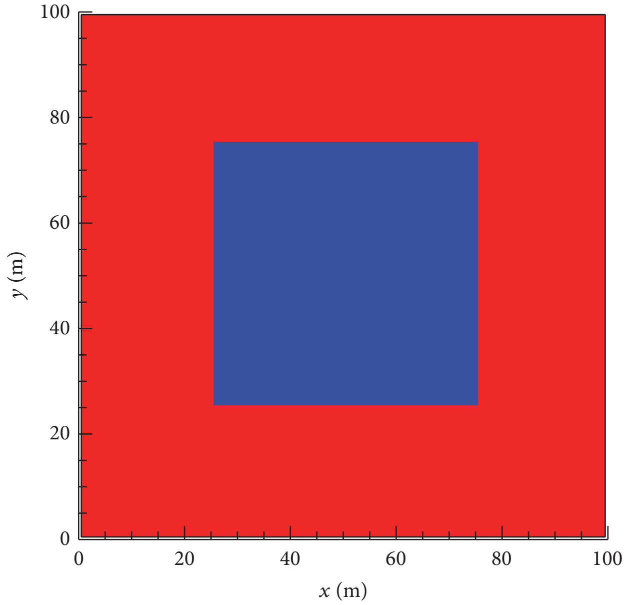


Figure 1: Computational domain and permeability.

3. Results and Discussions

The computational domain (100 m × 100 m) is shown in Figure 1. Permeability takes the value k_s in the blue area (50 m × 50 m in the center of the domain) and k_l in the red area. Natural gas with the main component of methane (CH_4) is the most important gas in subsurface reservoirs. Thus, it is used in the numerical cases in this paper. Initial pressure p_0 is given in the whole domain with zero injection or production. p_{lb} and p_{rb} are imposed on the left and the right boundaries. No flow boundary condition is applied to the top and bottom boundaries. All parameters are listed in Table 1, where 1 md = 9.869233×10^{-16} m^2.

In this condition, governing equations (see (4) and (7)~(10)) are solved to collect samples at different moments, using the highly accurate finite difference method (FDM) to ensure the grid-independent results for the grid number more than 60 × 60 [29]. Thus, the grid number in this paper (100 × 100) is enough. The temporal scheme is an explicit advancement for each time step ($\Delta t = 1.296$ s) with 2×10^6 time steps. After the total time scale of the simulation of 30 days, 2000 samples are collected. To evaluate the precision and computational speed of the POD model, the relative deviation ε and the acceleration ratio r are defined in the following:

$$\varepsilon = \frac{\left\| p_{\mathrm{POD}} - p_{\mathrm{FDM}} \right\|}{\left\| p_{\mathrm{FDM}} \right\|} \times 100\%, \tag{35}$$

$$r = \frac{t_{\mathrm{FDM}}}{t_{\mathrm{POD}}}, \tag{36}$$

where t is the CPU time for the computation and the subscripts "POD" and "FDM" represent the results from the POD model and from the direct calculation of the governing equations via the FDM. If $r > 1$, the POD model can accelerate the computation. The larger r represents the larger acceleration.

Table 1: Computational parameters.

Parameter	Value	Unit
ϕ	0.2	/
p_0	101325	Pa
p_{lb}	1013250	Pa
p_{rb}	101325	Pa
q	0	kg/(m^3·s)
k_s	1	md
k_l	100	md
R	8.3147295	J/(mol·K)
T	298	K
W	16×10^{-3}	Kg/mol
μ	11.067×10^{-6}	Pa·s
a	0.2283	Pa·m^6·mol^{-2}
b	4.278×10^{-5}	m^3/mol
M	2000	/
nx	100	/
ny	100	/
l_x	100	m
l_y	100	m
Δx	1	m
Δy	1	m
Δt	1.296	s
Simulation time scope	30	Days

3.1. POD Results for Ideal Gas.

First of all, the POD model is examined in the case of ideal gas ($Z \equiv 1$), where a and b in the Van der Waals equation (see (4)) are all zero. Thus, the expression of the gas compressibility (see (7)) decays to $c_g = 1/p$ so that (31) and (33) can be simplified as

$$\begin{aligned} HU_{m,n} &= \sum_{j=1}^{ny}\sum_{i=1}^{nx} \left(\varphi_m \varphi_n\right)_{i,j} \Delta x \Delta y, \\ HS_m &= \sum_{j=1}^{ny}\sum_{i=1}^{nx} \left(\varphi_m q\right)_{i,j} \Delta x \Delta y. \end{aligned} \tag{37}$$

As shown in Figure 2, the first POD mode occupies the vast majority (96.86%) of the total energy. This indicates that the first mode captures the main characteristics of the whole transient process. However, the relative deviation of the reconstruction results largely fluctuates with time. The maximum deviation is as high as 210% while the minimum deviation is only 4.6×10^{-2}% (Table 2). This phenomenon indicates that a dominant mode occupying the vast majority of energy can only obtain high-accurate reconstruction at some moments but loses the fidelity at other moments. To promote the reconstruction precision of the whole transient process, more POD modes should be included. To ensure high acceleration of the reduced-order model, the inclusion of the POD modes should be as little as possible. Thus, there

Table 2: Reconstruction deviations using top 1~top 7 modes.

POD modes	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7
$\varepsilon_{\max}$	210%	84%	46%	29%	27%	12%	13%
$\varepsilon_{\min}$	4.6×10^{-2}%	4.3×10^{-3}%	2.4×10^{-3}%	1.9×10^{-4}%	5.7×10^{-3}%	4.7×10^{-3}%	6.9×10^{-4}%

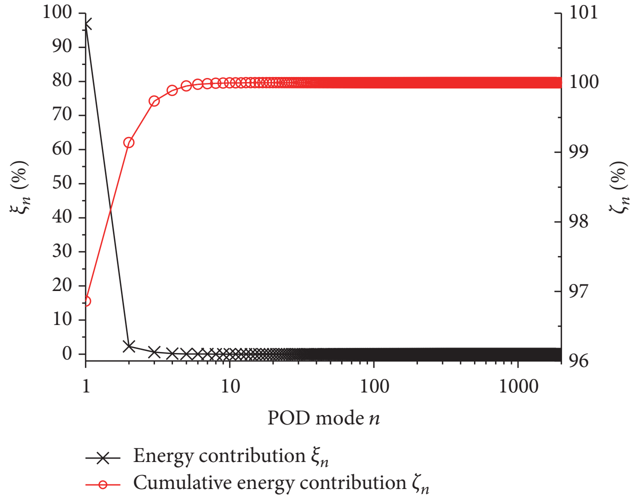


Figure 2: Energy spectrum of the POD modes.

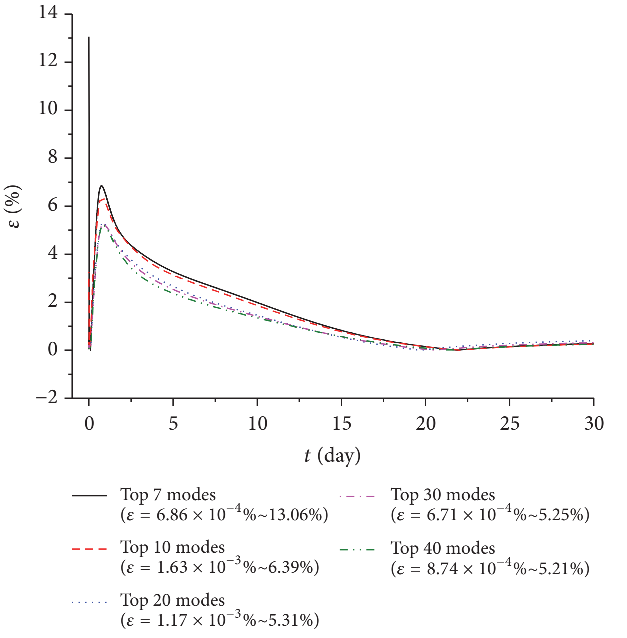
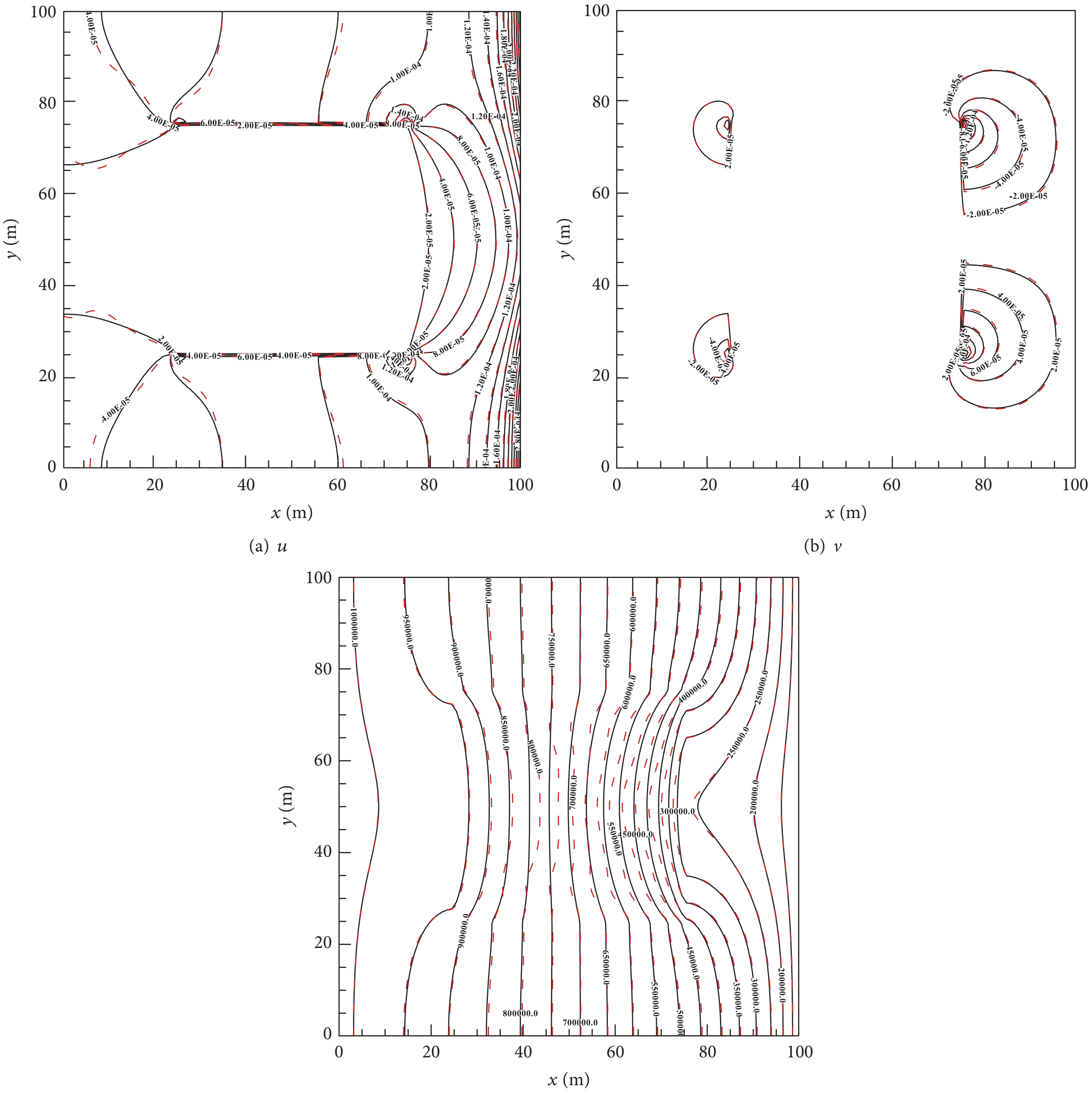


Figure 3: Reconstruction deviations using different numbers of POD modes.

is an optimal truncation number of modes. Theoretically, the number of the POD modes can be usually determined when the cumulative energy contribution achieves 100%. As shown in Figure 2, the cumulative energy contribution ζ_n achieves 100% at and after the 7th POD mode. Although the maximum deviation decreases rapidly with increasing number of POD modes, the maximum deviation is still as high as 13% when the top 7 modes are used. Therefore, more modes should be considered in the reduced-order model.

Relative deviations along with time, using more POD modes, are shown in Figure 3. With the increasing number of POD modes, the deviation converges rapidly when more than top 10 modes are used. The maximum deviation decreases greatly from 13.06% at the top 7 modes to 6.39% at the top 10 modes. After the top 10 modes, the decrease of the maximum deviation becomes much slower (5.31% for the top 20 modes, 5.25% for the top 30 modes, and 5.21% for the top 40 modes). The minimum deviations are all small. Thus, it can be considered that the reconstruction results converge at the number of top 20 modes. However, the computational time of the POD model is increasing with increasing number of POD modes, causing the acceleration ratio to be largely influenced (Table 3). The largest drop of the acceleration ratio occurs between the top 10 modes and top 20 modes. More POD modes cause the dimension of the POD model to increase much faster so that the computation of a larger equation system consumes much more time. Therefore, the optimal number of POD modes needs a balance between the precision and the acceleration. Between the top 10 modes and top 20 modes, the precision is promoted a little (ε from 6.39% to 5.31%) but the acceleration ratio of the POD model decreases from 24 for the top 10 modes to 6.6 for the top 20 modes, so that the number of the top 10 modes is an appropriate optimal number of POD modes. From the comparison, it should be noted that the theoretical energy criterion is not enough to determine the optimal number of POD modes. The balance of precision and acceleration should be considered. This will include much more POD modes with very small energy contribution, but their actual contributions to precision are important.

It can also be seen from Figure 3 that the maximum deviation (6.39%) and the minimum deviation (1.63×10^{-3}%) occur at about 0.8 days and about 22 days when the top 10 modes are used in the POD model. The velocity field and pressure field obtained by the POD model are compared with those obtained by FDM to examine the reconstruction precision of local characteristics. From Figure 4, it is clear that the two components of Darcy velocity and pressure are all reconstructed well with tiny local deviation. From Figure 5, reconstructed velocity component v and reconstructed pressure agree well with those of FDM, but reconstructed velocity component u can only capture part of the features. Fortunately, this maximum deviation only exists in a very narrow range in Figure 3. The deviation at other moments decreases rapidly. We examine the local flow field at the mean deviation (3.54%) in Figure 6 and find that the local distributions of u, v, and p are well reconstructed. Thus, the POD model using the top 10 POD modes can reconstruct the flow field in high precision for the whole transient process.



(a) u

(b) v

(c) p

Figure 4: Flow field comparison for the minimum deviation. Black line: FDM; red dashed line: POD.

3.2. POD Results for Real Gas. For real gas, compressibility factor (Z) and compressibility (c_g) should be computed by the equation of state (see (4)) and the equation of real gas compressibility (see (7)). POD results for the same case in Section 3.1 are obtained by solving (34). The reconstruction precision ($\varepsilon = 2.78 \times 10^{-3}\%{\sim}6.39\%$) is as high as that of ideal gas (Figure 3). Flow field comparisons are also very similar to those in Figures 4–6. Thus, these results are not redundantly shown here. The above POD model maintains high precision for real gas simulation.

However, the acceleration ratio is as low as 0.9 (i.e., 6240 s/7256 s in Table 4), which means the POD model causes the simulation to be even slower than FDM. This is quite different from the high acceleration ratio for ideal gas ($r = 24$ in Table 3). The main difference between the computations of real gas and ideal gas is that the equation of state (EOS) should be computed for real gas. We analyze the different contributions of the total computational time for FDM and POD in Table 4 and find two important phenomena: (1) the largest part of the total computational time is mainly consumed on EOS for both FDM and POD (92% for FDM and 78% for POD); the EOS cannot be expressed as an explicit function of pressure so that it cannot be decomposed by the POD; it can only be calculated locally for both FDM and POD

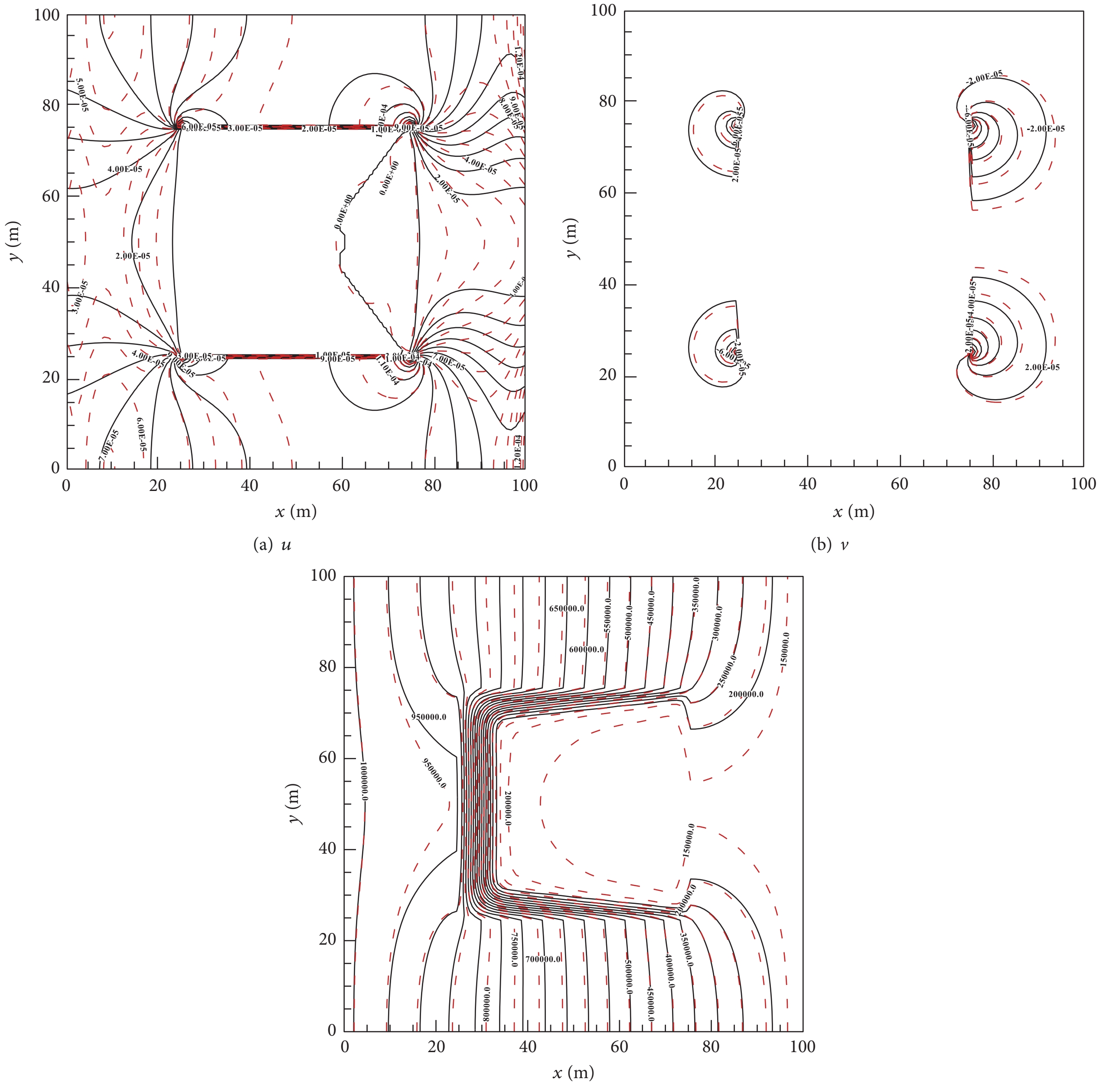
Table 3: Computational time using top 7~top 40 modes.

POD modes	Top 7	Top 10	Top 20	Top 30	Top 40
CPU time of POD	15 s	23 s	85 s	246 s	420 s
r	37	24	6.6	2.2	1.3

Note. The computational time of FDM is 560 s.

Table 4: Computational time analyses in the case of real gas.

	Total CPU time	CPU time for EOS	CPU time for flow equations	Time contribution of EOS	Time contribution of flow equation
FDM	6240 s	5747 s	493 s	92%	8%
POD	7256 s	5660 s	1596 s	78%	22%

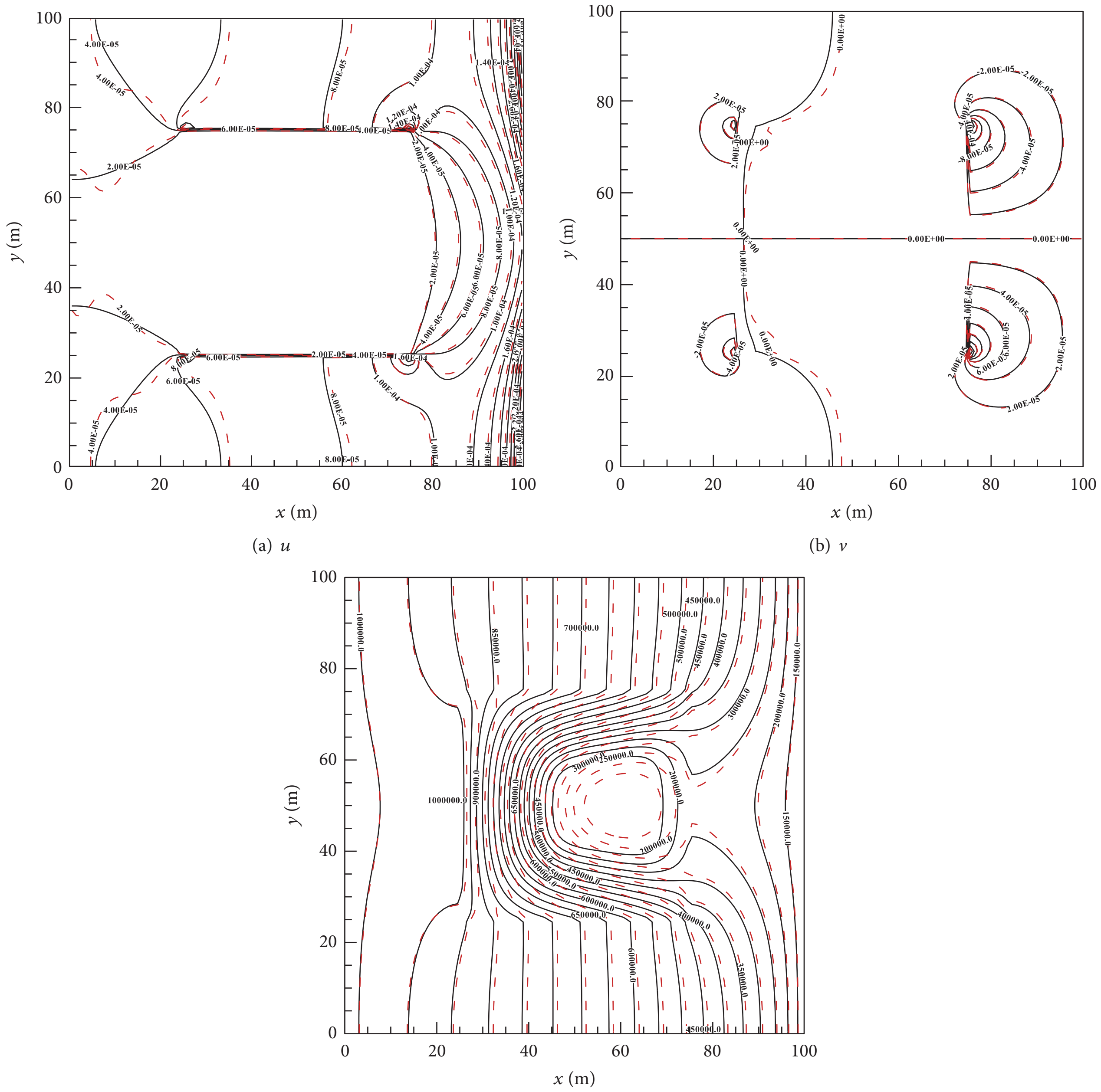


(a) u

(b) v

(c) p

Figure 5: Flow field comparison for the maximum deviation. Black line: FDM; red dashed line: POD.



(a) u

(b) v

(c) p

Figure 6: Flow field comparison for the mean deviation. Black line: FDM; red dashed line: POD.

and thus the main computational time of real gas simulation is not reduced in the POD model; (2) the computational time for flow equations in POD (1596 s) is much longer than that in FDM (493 s) with much higher contribution (22%) than that of FDM (8%). This means the POD model is still slower than FDM even not considering the computational time of EOS. This can be explained using the expression of the POD model. In the model (see (34)), all terms are constant for time advancement except the projection of the unsteady term (see (31)) and the projection of the source term (see (33)). They should be calculated in every time step because they contain the variables changing with time (c_g, p, and Z). This kind of calculation largely increases the total computations.

The above analyses indicate that the improvement of the acceleration ability of the POD model should reduce the time contribution of EOS and avoid the frequent computations of the two projection terms in the POD computation. According to these two points, we propose a new method: EOS is only calculated once at the initial time. Then, the initial compressibility, the initial compressibility factor, and the initial pressure are used to approximate the two projection terms in (31) and (33), which are treated as constants in the following transient computation of the POD model. Through this treatment, the calculations of EOS and the two projection terms in every time step are all avoided for the POD computation so that the computational time of the POD

Table 5: Computational time for real gas after improvement.

	Total CPU time	CPU time for EOS	CPU time for flow equations	Time contribution of EOS	Time contribution of flow equation
FDM	6240 s	5747 s	493 s	92%	8%
POD	23 s	0.02 s	22.98 s	0.09%	99.91%

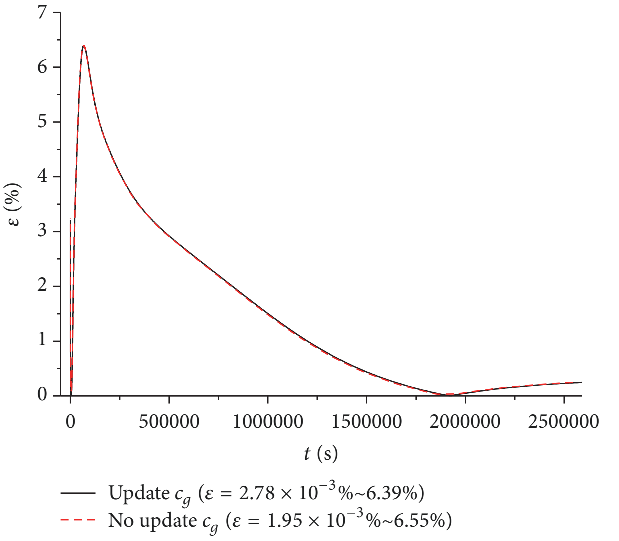


Figure 7: Precision of the POD model for different treatment of compressibility.

model is expected to be reduced greatly. The approximations of the two projection terms are shown as follows:

$$HU_{m,n} = \sum_{j=1}^{ny}\sum_{i=1}^{nx}\left(c_g^0 p^0 \varphi_m \varphi_n\right)_{i,j} \Delta x \Delta y,$$
$$HS_m = \sum_{j=1}^{ny}\sum_{i=1}^{nx}\left(\varphi_m Z^0 q\right)_{i,j} \Delta x \Delta y, \tag{38}$$

where c_g^0, Z^0, and p^0 are the compressibility, the compressibility factor, and the pressure in the initial condition ($t = 0$), respectively.

Figure 7 shows that the relative deviation of the above treatment is almost the same as that updating the compressibility every time step, indicating that this treatment does not affect the precision of the POD model. Table 5 shows that the total computational time of the POD model has been largely reduced because the computational time of EOS and flow equations are all reduced greatly. The time decrease of EOS is because the EOS is only calculated once. The time decrease of flow equations is because the two projection terms are not calculated with time advancement of the POD model. These results demonstrate that large acceleration ratio (r = 6240 s/23 s = 271) of POD for real gas simulation can only be achieved when the time contribution of EOS is greatly reduced.

3.3. Verification of the Improved POD Model in a More Realistic Case. The improved POD model in the above section is applied to a more complex case of real gas flow simulation to confirm its capability. Permeability field is shown in Figure 8, where the red area has a permeability 100 md and the blue area has a permeability 1 md. The large permeability represents the main flow path of the gas in the reservoir such as soil or sand. The small permeability represents the main obstacles such as rock. Higher boundary pressure on the left border (100 atm) is used because gas pressure is usually high in engineering. Other parameters are the same as the previous case.

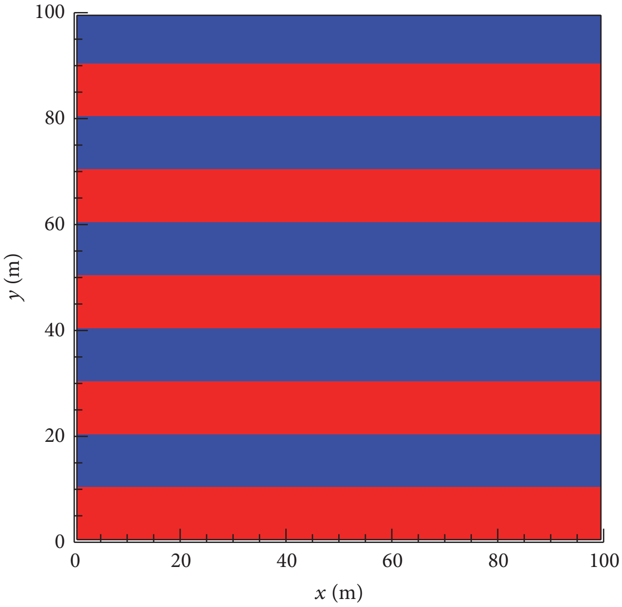


Figure 8: Real gas flow in a more realistic reservoir.

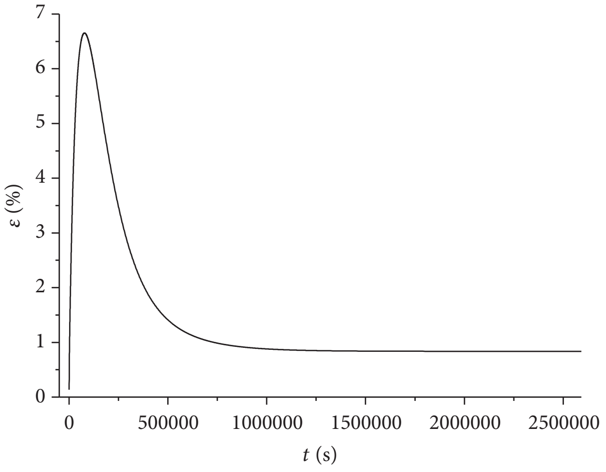
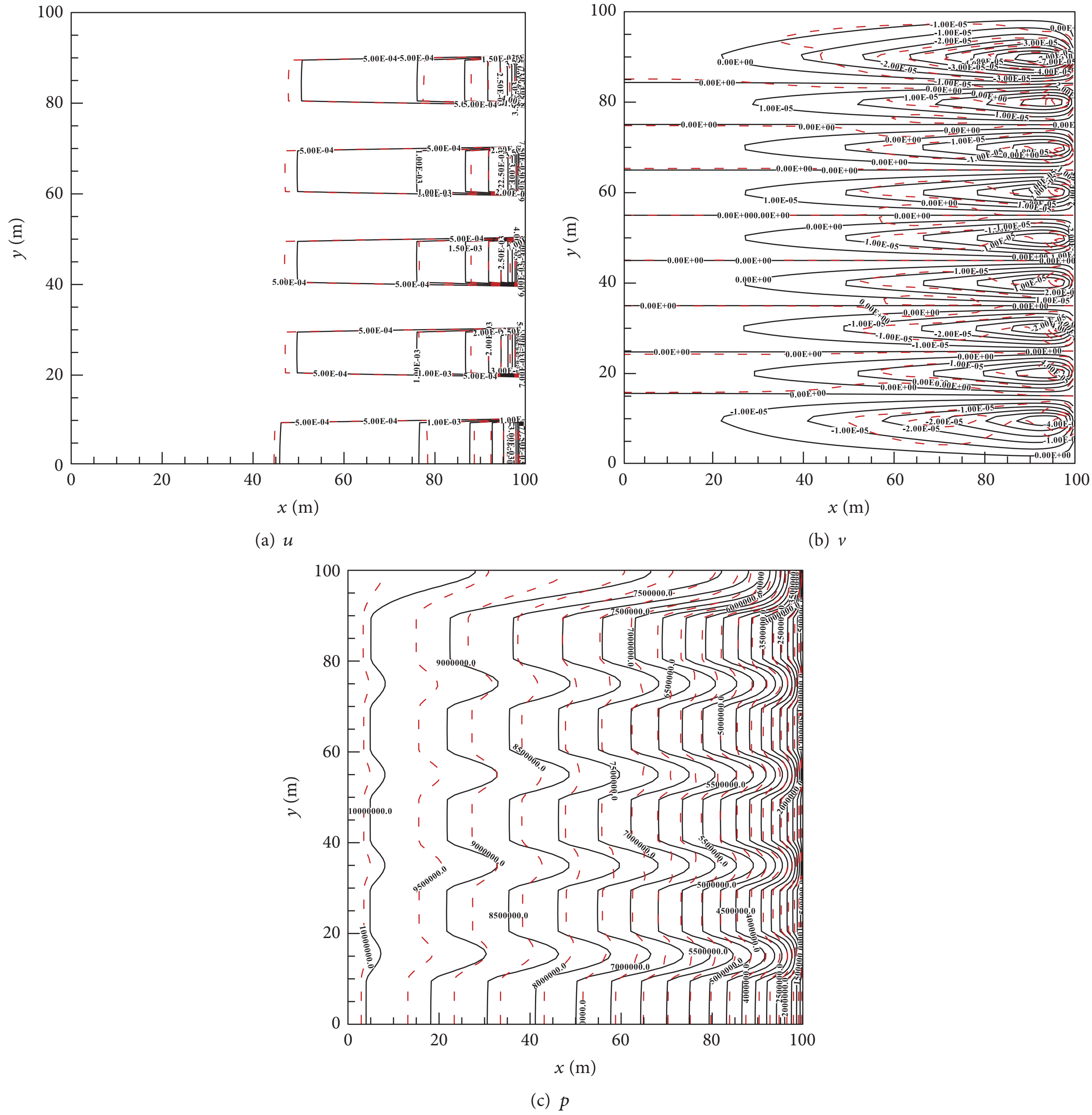


Figure 9: Precision of the improved POD model.

As shown in Figure 9, the relative deviations are also as low as the previous case. The maximum and minimum



(a) *u*

(b) *v*

(c) *p*

Figure 10: Flow field comparison at the maximum deviation for real gas. Black line: FDM; red dashed line: POD.

deviations are 6.65% and 0.13%, respectively, indicating high precision of the improved POD model. It is further confirmed in Figures 10 and 11 that the local flow fields are reconstructed well at the maximum and minimum deviations. A little larger deviation of pressure in Figure 10(c) may be caused by the approximation of gas compressibility. However, the main features of the pressure field are correct while the velocity field has much smaller deviation. Thus, the precision is acceptable.

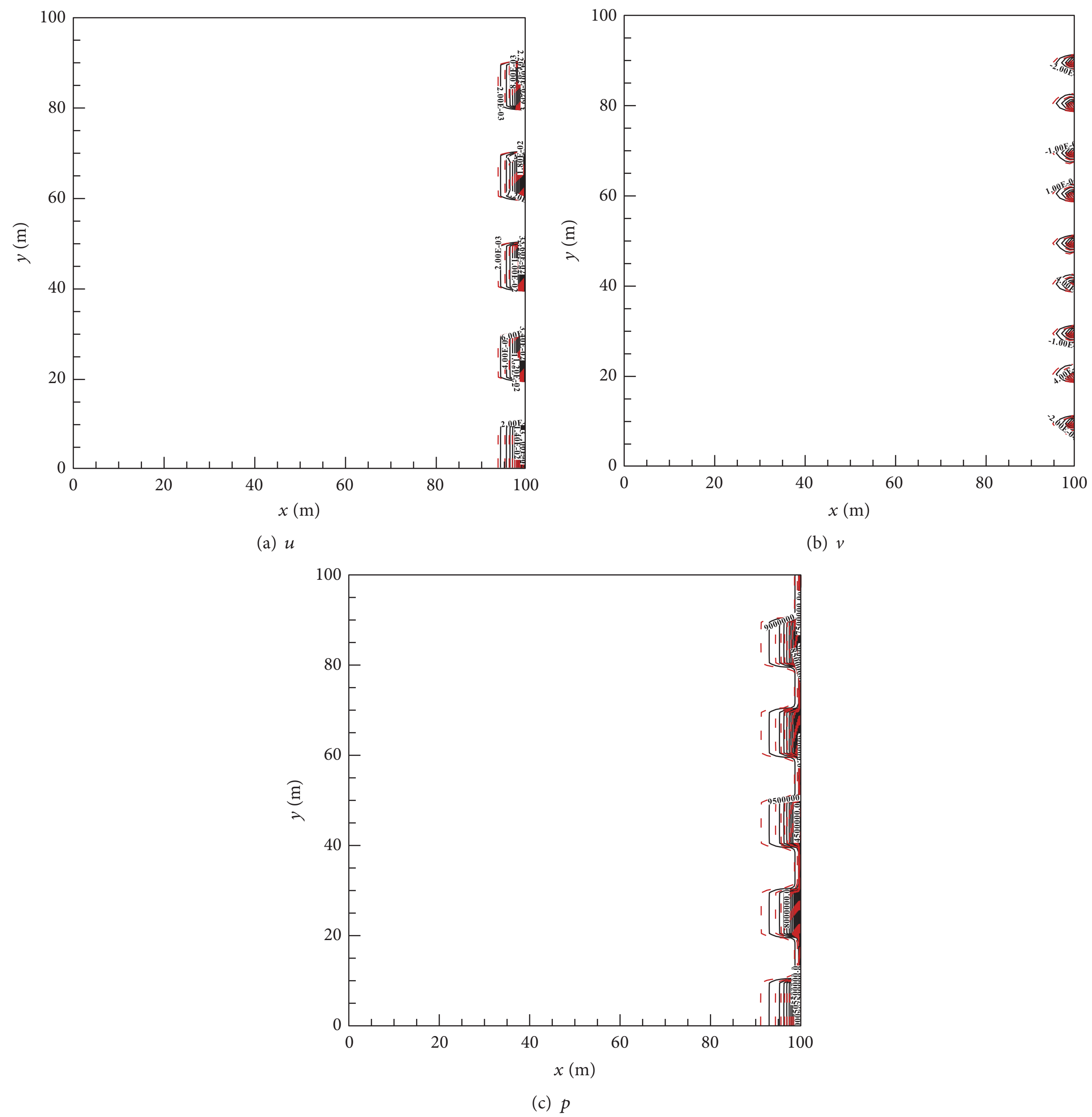
Table 6 shows that the computational time is reduced from 5823 s for FDM to 25 s for POD. The acceleration ratio is as high as 233. Therefore, the improved POD model can still greatly save computational time for more complex reservoir and higher pressure.

Table 6: Computational time comparison for real gas.

CPU time of FDM	CPU time of POD	*r*
5823 s	25 s	233

4. Conclusions

Proper orthogonal decomposition is utilized in gas reservoir simulation to accelerate the simulation speed of gas flow in single-continuum porous media. High-precision reconstruction can be achieved using only 10 POD modes with the deviations as low as 1.63×10^{-3}%~6.39% for ideal gas and



(a) u

(b) v

(c) p

Figure 11: Flow field comparison at the minimum deviation for real gas. Black line: FDM; red dashed line: POD.

1.95×10^{-3}%~6.55% for real gas. The acceleration ratio is as high as 24 for the typical POD model of the ideal gas flow. However, the computational speed of the typical POD model of the real gas flow is even slower than FDM. Two key points for improving the computational speed of the POD model are discussed and verified:

(1) The computation of EOS should be avoided in the solving process of the POD model because the total computational time is dominated by EOS.

(2) POD projection terms containing compressibility of gas should not be updated in every time step. Otherwise, the computational time of flow equations will also be longer than FDM.

According to the two points, we proposed a new method to approximate the projection terms in all time steps using the initial compressibility so that EOS only needs to be calculated once at the initial condition. After this treatment, it is verified in two different cases that the computational speed of the POD model is largely promoted while high precision is retained (0.13%~6.65%). The computational time of the real gas reservoir simulation is reduced from 6240 s and 5823 s of FDM to 23 s and 25 s of POD for the two cases. The acceleration ratios are 271 and 233, respectively.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The work presented in this paper has been supported by the National Natural Science Foundation of China (NSFC) (nos. 51576210, 51325603, and 51476073) and the Science Foundation of China University of Petroleum, Beijing (nos. 2462015BJB03, 2462015YQ0409, and C201602). This work is also supported by the Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality (no. IDHT20170507) and the Foundation of Key Laboratory of Thermo-Fluid Science and Engineering (Xi'an Jiaotong University), Ministry of Education, Xi'an 710049, China (KLTFSE2015KF01).

References

[1] H. Darcy, *Les Fontaines Publiques de la Vill de Dijon*, Dalmont, Paris, France, 1856.

[2] Z. Chen, G. Huan, and Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2006.

[3] J. Chen and Z. Chen, "Three-dimensional superconvergent gradient recovery on tetrahedral meshes," *International Journal for Numerical Methods in Engineering*, vol. 108, no. 8, pp. 819–838, 2016.

[4] J. Moortgat, S. Sun, and A. Firoozabadi, "Compositional modeling of three-phase flow with gravity using higher-order finite element methods," *Water Resources Research*, vol. 47, no. 5, Article ID W05511, pp. 1–26, 2011.

[5] S. Sun and J. Liu, "A locally conservative finite element method based on piecewise constant enrichment of the continuous galerkin method," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2528–2548, 2009.

[6] S. Sun and M. F. Wheeler, "Discontinuous Galerkin methods for simulating bioreactive transport of viruses in porous media," *Advances in Water Resources*, vol. 30, no. 6-7, pp. 1696–1710, 2007.

[7] S. Sun and M. F. Wheeler, "Analysis of discontinuous Galerkin methods for multicomponent reactive transport problems," *Computers & Mathematics with Applications. An International Journal*, vol. 52, no. 5, pp. 637–650, 2006.

[8] S. Sun and M. F. Wheeler, "A dynamic, adaptive, locally conservative, and nonconforming solution strategy for transport phenomena in chemical engineering," *Chemical Engineering Communications*, vol. 193, no. 12, pp. 1527–1545, 2006.

[9] L. Luo, B. Yu, J. Cai, and X. Zeng, "Numerical simulation of tortuosity for fluid flow in two-dimensional pore fractal models of porous media," *Fractals*, vol. 22, no. 4, Article ID 1450015, 2014.

[10] W. Wei, J. Cai, X. Hu et al., "A numerical study on fractal dimensions of current streamlines in two-dimensional and three-dimensional pore fractal models of porous media," *Fractals*, vol. 23, no. 1, Article ID 1540012, 2015.

[11] V. Chandra, A. Barnett, P. Corbett et al., "Effective integration of reservoir rock-typing and simulation using near-wellbore upscaling," *Marine and Petroleum Geology*, vol. 67, pp. 307–326, 2015.

[12] V. Gholami and S. D. Mohaghegh, "Fuzzy upscaling in reservoir simulation: An improved alternative to conventional techniques," *Journal of Natural Gas Science and Engineering*, vol. 3, no. 6, pp. 706–715, 2011.

[13] M. G. Correia, C. Maschio, J. C. von Hohendorff Filho, and D. J. Schiozer, "The impact of time-dependent matrix-fracture fluid transfer in upscaling match procedures," *Journal of Petroleum Science and Engineering*, vol. 146, pp. 752–763, 2016.

[14] S. Gholinezhad, S. Jamshidi, and A. Hajizadeh, "Quad-tree decomposition method for areal upscaling of heterogeneous reservoirs: application to arbitrary shaped reservoirs," *Fuel*, vol. 139, pp. 659–670, 2015.

[15] A. Q. Raeini, M. J. Blunt, and B. Bijeljic, "Direct simulations of two-phase flow on micro-CT images of porous media and upscaling of pore-scale forces," *Advances in Water Resources*, vol. 74, pp. 116–126, 2014.

[16] L. S. K. Fung, M. O. Sindi, and A. H. Dogru, "Multiparadigm parallel acceleration for reservoir simulation," *SPE Journal*, vol. 19, no. 4, pp. 716–725, 2014.

[17] G. Yu, B. Yu, D. Han, and L. Wang, "Unsteady-state thermal calculation of buried oil pipeline using a proper orthogonal decomposition reduced-order model," *Applied Thermal Engineering*, vol. 51, no. 1-2, pp. 177–189, 2013.

[18] D. Han, B. Yu, Y. Wang, Y. Zhao, and G. Yu, "Fast thermal simulation of a heated crude oil pipeline with a BFC-Based POD reduced-order model," *Applied Thermal Engineering*, vol. 88, pp. 217–229, 2015.

[19] Y. Wang, B. Yu, X. Wu, and P. Wang, "POD and wavelet analyses on the flow structures of a polymer drag-reducing flow based on DNS data," *International Journal of Heat and Mass Transfer*, vol. 55, no. 17-18, pp. 4849–4861, 2012.

[20] Y. Wang, B. Yu, Z. Cao, W. Zou, and G. Yu, "A comparative study of POD interpolation and POD projection methods for fast and accurate prediction of heat transfer problems," *International Journal of Heat and Mass Transfer*, vol. 55, no. 17-18, pp. 4827–4836, 2012.

[21] Y. Wang, B. Yu, and S. Sun, "Fast prediction method for steady-state heat convection," *Chemical Engineering & Technology*, vol. 35, no. 4, pp. 668–678, 2012.

[22] M. Ghommem, V. M. Calo, and Y. Efendiev, "Mode decomposition methods for flows in high-contrast porous media: a global approach," *Journal of Computational Physics*, vol. 257, pp. 400–413, 2014.

[23] M. Ghommem, M. Presho, V. M. Calo, and Y. Efendiev, "Mode decomposition methods for flows in high-contrast porous media: global-local approach," *Journal of Computational Physics*, vol. 253, pp. 226–238, 2013.

[24] Y. Efendiev, J. Galvis, and E. Gildin, "Local-global multiscale model reduction for flows in high-contrast heterogeneous media," *Journal of Computational Physics*, vol. 231, no. 24, pp. 8100–8113, 2012.

[25] M. J. Patel, E. F. May, and M. L. Johns, "High-fidelity reservoir simulations of enhanced gas recovery with supercritical CO_2," *Energy*, vol. 111, pp. 548–559, 2016.

[26] W. Shen, Y. Xu, X. Li, W. Huang, and J. Gu, "Numerical simulation of gas and water flow mechanism in hydraulically fractured shale gas reservoirs," *Journal of Natural Gas Science and Engineering*, vol. 35, pp. 726–735, 2016.

[27] X. Peng, M. Wang, Z. Du, B. Liang, and F. Mo, "Upscaling and simulation of composite gas reservoirs with thin-interbedded shale and tight sandstone," *Journal of Natural Gas Science and Engineering*, vol. 33, pp. 854–866, 2016.

[28] D. A. Wood, W. Wang, and B. Yuan, "Advanced numerical simulation technology enabling the analytical and semi-analytical modeling of natural gas reservoirs," *Journal of Natural Gas Science and Engineering*, vol. 26, pp. 1442–1451, 2015.

[29] Y. Wang and S. Sun, "Direct calculation of permeability by high-accurate finite difference and numerical integration methods," *Communications in Computational Physics*, vol. 20, no. 2, pp. 405–440, 2016.