GPU-based Real-time Porous Medium Dynamic Network Simulation

Chenxi Li, Min Li and Yanbing Tang

ABSTRACT

Porous medium network simulation is to accurately describe the seepage law of multiphase fluids in porous media. Researchers have studied the relationship between various macroscopic physical properties of multiphase fluids (e.g. relative permeability, capillary pressure and fluid saturation) from theoretical, experimental, and computer simulation. This research introduces the construction method of stochastic pore network model and describes the calculation steps of pore network pressure field. In this research, the computational efficiency of the GPU-based pore network was compared with CPU. The calculation speed and simulation model scale of the pressure domain of the random pore network were improved.

1. INTRODUCTION

In the study of the microscopic pore-throat structure of tight oil, it is necessary to construct a pore network model corresponding to the tight oil reservoir. This model is used to dynamically simulate the oil-water two-phase percolation process of tight oil reservoirs under different water flooding conditions, wettability and fluid properties [1].

At present, it has been possible to obtain two-phase percolation flow patterns, relative permeability, oil saturation and fluid distribution characteristics under
different displacement conditions, injection multiples, and displacement times by simulation. However, there are still problems in the current calculation methods that need to be improve.

2. THEORIES

2.1 Generation of stochastic pore network model

Taking the two-dimensional square mesh network model as an example, this paper introduces the modeling method of percolation network model. The percolation network model is characterized by the shape rules of the model, and the position of the nodes are fixed. Take a two-dimensional uniform square network with a node number 5 × 5 as an example, shown in Figure 1.

1) Determine the size and number of nodes of the network model. 2) Calculate the coordinates of each node in the network model. 3) Using a random method, determine whether there is tube bundle connection between adjacent nodes in the x direction and the y direction (50% in the x direction and 70% in the y direction).

![Figure 1. Example of percolation network model construction.](image)

2.2 Pressure field simulation of network model

The difference between the inflow and outflow at any network node over a period of time is always equal to 0[2], denoted as:

\[
\sum_{j=1}^{z_i} Q_{i,j} = 0
\]

(1)
Once the pressure on all nodes in the network is obtained, the fluid volume of the network input and output can be calculated. In this research, the direct solving method was used [3].

By performing a pressure conservation calculation on each node in the network model, a linear equation group equal to the number of pressure nodes of the unknown node can be obtained.

1) Assign an initial value to each node in the network; 2) Establish a relational expression between pressure and flow between nodes.

\[
\begin{align*}
0(i, j, k) & = 1(i - 1, j, k) + 6(i, j, k + 1) + 3(i, j - 1, k) + 2(i + 1, j, k) + x + y + z + 4(i, j + 1, k) + 5(i, j, k - 1)
\end{align*}
\]

Figure 2. Schematic diagram of simple cubic network node unit.

This paper uses a simple two-dimensional pore network as an example, as shown in Figure 3.

The pressure conservation formula is deformed to obtain a linear equation of the form \( Ax = B \).
\[ \sum_{i} q_{ij} = 0 \quad \Rightarrow \quad [G] \times [p] = [q] \]

\[
\begin{bmatrix}
g_{1} & -g_{12} & 0 & -g_{14} & 0 & 0 \\
-g_{12} & g_{2} & -g_{23} & 0 & 0 & 0 \\
0 & -g_{23} & g_{3} & -g_{34} & 0 & -g_{36} \\
-g_{14} & 0 & -g_{34} & g_{4} & -g_{45} & 0 \\
0 & 0 & 0 & -g_{45} & g_{5} & -g_{56} \\
0 & 0 & 0 & 0 & -g_{56} & g_{6} \\
\end{bmatrix}
\begin{bmatrix}
p_{1} \\
p_{2} \\
p_{3} \\
p_{4} \\
p_{5} \\
p_{6} \\
\end{bmatrix}
= 
\begin{bmatrix}
q_{01} \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

\( g_{ij} \) is the conductivity of the current network connection; \( p_{i} \) is the pressure of each network node; \( q_{ij} \) is the pressure of the network connection.

Solving this matrix can get the pressure of each node in the network, and then can calculate the flow at the inlet and outlet of the model.

### 2.3 Dynamic network computing based on GPU

In order to calculate the pressure field more effectively, this paper used several matrix solving libraries such as Eigen, PARALUTION, AMGCL, Cusp, ViennaCL and AmgX for comparison.

The experiment uses Eigen's running time as a reference to compare the efficiency of each GPU line calculation library. The benchmark platform we used is shown in Table II.

---

**TABLE I. COMPARISON OF MATRIX LIBRARIES.**

<table>
<thead>
<tr>
<th>Library</th>
<th>Cross-platform</th>
<th>OpenMP</th>
<th>CUDA</th>
<th>OpenCL</th>
<th>Stable release</th>
<th>CUDA version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigen</td>
<td>●</td>
<td>●</td>
<td>/</td>
<td>/</td>
<td>2018.12</td>
<td>/</td>
</tr>
<tr>
<td>PARALUTION</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>2016.1</td>
<td>CUDA8</td>
</tr>
<tr>
<td>AMGCL</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>2018.5</td>
<td>CUDA9</td>
</tr>
<tr>
<td>Cusp</td>
<td>●</td>
<td>/</td>
<td>●</td>
<td>/</td>
<td>2014.10</td>
<td>CUDA7</td>
</tr>
<tr>
<td>ViennaCL</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>2016.1</td>
<td>CUDA8</td>
</tr>
<tr>
<td>AmgX</td>
<td>○²</td>
<td>○³</td>
<td>●</td>
<td>/</td>
<td>2018.2</td>
<td>CUDA10</td>
</tr>
</tbody>
</table>

1. The Cross-platform in this table refer to Windows, Linux and macOS systems;
2. AmgX only supports Windows and Linux systems
3. AmgX only supports OpenMP during data initialization
### TABLE II. COMPUTING PLATFORM.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel i7-7700HQ/Intel E5 2680 v3 x 2</td>
</tr>
<tr>
<td>GPU</td>
<td>GTX 1060M/GTX Titan(Kepler)</td>
</tr>
<tr>
<td>OS</td>
<td>Windows 10 1809</td>
</tr>
<tr>
<td>Development language</td>
<td>C/Python/MATLAB</td>
</tr>
</tbody>
</table>

### 3. RESULTS

The experiments in this paper used a pore network of about 2 million nodes (the number of network pores is $10^6 * 10^3 * 10$). The calculation efficiency is based on the CPU operation time. The experimental results are shown in Table III and Figure 4:

### TABLE III. Runtimes comparison of each computing library.

<table>
<thead>
<tr>
<th>Library</th>
<th>Single iteration time (sec)</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigen</td>
<td>126.472</td>
<td>100%</td>
</tr>
<tr>
<td>PARALUTION</td>
<td>3.174</td>
<td>3049%</td>
</tr>
<tr>
<td>AMGCL</td>
<td>3.856</td>
<td>2507%</td>
</tr>
<tr>
<td>Cusp</td>
<td>4.696</td>
<td>2693%</td>
</tr>
<tr>
<td>ViennaCL</td>
<td>3.671</td>
<td>3445%</td>
</tr>
<tr>
<td>AmgX</td>
<td>1.561</td>
<td>8102%</td>
</tr>
</tbody>
</table>

![Figure 4. Efficiency comparison of each computing library.](image)
As can be seen from the results above, NVIDIA’s AmgX library is superior to other matrix library in solving performance. Meanwhile, AmgX using multi-layer matrix decomposition to realize and solve large-scale matrix solution\[4\]. In the recent experiments, the matrix has been expanded to a level of 10 million and speedup to 100 times faster than CPU.

4. CONCLUSIONS

The AmgX library based on CUDA platform accelerates the calculation of the pressure field in the pore network simulation operation. At the same time, the high efficiency and ease of use of AmgX not only directly assists reservoir simulation studies, but also supports in subsequent deep learning reservoir simulation studies.

REFERENCES

