Parallel Simulation of Wormhole Propagation with the Darcy-Brinkman-Forchheimer Framework

Yuanqing WU* Amgad SALAMA, Shuyu SUN

Computational Transport Phenomena Laboratory (CTPL), Division of Physical Sciences and Engineering (PSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

*Corresponding author

E-mail: Yuanqing.Wu@kaust.edu.sa, wuyuanq@gmail.com.

Telephone: 00966568856349, 008613928772190
Abstract

The acid treatment of carbonate reservoirs is a widely practiced oil and gas well stimulation technique. The injected acid dissolves the material near the wellbore and creates flow channels that establish a good connectivity between the reservoir and the well. Such flow channels are called wormholes. Different from the traditional simulation technology relying on Darcy framework, the new Darcy-Brinkman-Forchheimer (DBF) framework is introduced to simulate the wormhole forming procedure. The DBF framework considers both large and small porosity conditions and should output better simulation results than the Darcy framework. To process the huge quantity of cells in the simulation grid and shorten the long simulation time of the traditional serial code, a parallel code with FORTRAN 90 and MPI was developed. The experimenting field approach to set coefficients in the model equations was also introduced. Moreover, a procedure to fill in the coefficient matrix in the linear system in the solver was described. After this, 2D dissolution experiments were carried out. In the experiments, different configurations of wormholes and a series of properties simulated by both frameworks were compared. We conclude that the numerical results of the DBF framework are more like wormholes and more stable than the Darcy framework, which is a demonstration of the advantages of the DBF framework. Finally, the scalability of the parallel code was evaluated, and we conclude that superlinear scalability can be achieved.

Keywords: wormhole, Darcy-Brinkman-Forchheimer framework, parallelism
1. Introduction

A wormhole is a phenomenon in which a worm-like hole is generated and propagated in the subsurface formations due to the injection of acids into a supercritical acid dissolution system. Although acid dissolution fronts can be generated in both subcritical and supercritical acid dissolution systems, they are only unstable in supercritical acid dissolution systems, so that wormholes can only be generated and propagated in such supercritical acid dissolution systems [1-4]. Engineers and hydrologists appeal to this technique when the efficiency of oil production wells deteriorates due to the deposition of mud and fines at the perforated well pore pipe. In this case, acids are injected into the formations to promote chemical reactions. Such reactions cause the dissolution of deposits and propagate areas around the well pore with larger void spaces to facilitate the displacement of fluids. The advancement of the chemical reaction front does not propagate uniformly in the injection direction. The heterogeneity of porosity and permeability in the subsurface formations is an essential feature that promotes the non-uniformity of the chemical reaction front. The chemical reaction front tends to advance in certain directions more than other directions, and a wormhole pattern is established. Because of its importance in subsurface reservoir management, theoretical, experimental and numerical studies of wormhole generation and propagation in carbonate rocks have been a topic of key interest for research in the last forty years.

Theoretically, wormhole generation and propagation belong to the chemical dissolution-front instability problems in porous rocks. Because physical and chemical dissolution-front instability problems are closely associated with the geosciences [5-12],
geo-environmental engineering [13-17] and petroleum engineering [1-3], Zhao and his co-workers have conducted extensive and systematic theoretical studies in recent years [1-19]. In particular, many important factors such as the mineral reactive surface area [6], mineral dissolution ratio [8], solute dispersion [9], medium anisotropy [12], temperature effect [18,19], and medium and fluid compressibility [10] have been successfully considered in both theoretical analyses and computational simulations. Due to the importance of this type of work, it has become an essential part of the emerging field of computational geosciences [4]. The major conclusions of the existing theoretical work are (1) wormholes can only be generated and propagated in a supercritical acid dissolution system in which the Zhao number is greater than the corresponding critical Zhao number of the system; (2) wormholes may have different modes, such as the fundamental mode, the fingering mode and fractal mode, depending on the Zhao number of the supercritical acid dissolution system; (3) by comparing the Zhao number of an acid dissolution system with the corresponding critical Zhao number of the acid dissolution system, the instability of the acid dissolution front in carbonate rocks can be assessed [1-3].

Experimentally, it is important to develop methods to characterise wormhole structures. In this regard, a number of techniques have been implemented including the injection of molten metals that have a lower melting point (e.g., Wood’s metal) into the acid-etched channels of the porous matrix and allowing the metal to be solidified. Then, the complete dissolution of the porous core in acid leaves the solid metal casting [20]. However, the delicate branches of wormholes tend to break during the dissolution process. Moreover, neutron radiography has been recently used to image the core using the advantage that a cadmium component (which has a high cross section compared to that of
the surrounding porous media) exists in the Wood’s metal. Another technique is to use neutron transmission tomography to generate patterns of porosity variations [21]. Alternatively, many researchers have considered investigating different processes involved in the dynamics of wormhole propagation. For example, Qiu et al. [22] investigated the effects of the diffusion coefficients of HCl acid as it reacts with calcite. Sayed et al. [23] investigated the effect of the presence of crude oil in the formation on the performance of emulsified acid in stimulating carbonate formations. Kumar et al. [24] demonstrated the impact of oil saturation on wormholing characteristics while acidizing field and outcrop cores under reservoir conditions.

Numerically, a number of modelling approaches have been developed to investigate the propagation dynamics of wormholes. This includes the capillary tube model, network model and two-scale model. Although the capillary tube model [25] can describe the effect of fluid leakage and the transport and reaction mechanisms in the wormholes, it cannot provide us with the conditions that cause the formation of wormholes. The network model was developed by Hoefner and Fogler [20], Fredd and Fogler [26], and Daccord et al. [27]. This model uses a network of tubes to represent a porous medium. The acid reacts at the wall of the tube, and dissolution is developed, resulting in an increase in the tube radius. Although the network model can predict the dissolution patterns, the results are very different from the experimental results. In addition, a core size simulation using the network model is computationally very expensive. A two-scale model, averaged model or continuum model [28-30] can describe the dissolution patterns based on both the Darcy scale and pore scale. The information achieved by semi-empirical equations on the pore scale is passed to the Darcy scale equations to describe
the dissolution procedure. The development of dissolution changes the pore structures, which also changes the property parameters such as the porosity in the pore scale model. Therefore, the information on both the Darcy scale and pore scale can communicate with each other in the whole simulation procedure. Based on the two-scale model, much work has been carried out. Liu et al. [31] studied the wormhole propagation behaviour in the invaded zone and compressed zone, concluding that the effect of the compressed zone on the wormhole propagation increases with the decrease of the compressibility factor. Liu et al. [32] also studied the effect of the two distribution laws of porosity on the dissolution patterns and concluded that the results from the normal distribution law are closer to the experimental observations than those from the uniform distribution law. Different from much of the two-scale model study that has only 2D simulation, Maheshwari et al. [33] extended the 2D continuum model to the 3D case. Based on the 3D case, they studied the effect of the acid-injection rate on the amount of acid required to break through. They also studied the flow dynamics inside a wormhole. In addition to the three popular models, another modelling methodology based on the continuum hypothesis as applied to porous media has been suggested. In this framework, field variables represent continuous functions of space and time, and the governing conservation equations are described in the form of differential equations [34-36]. Zhao et al. [3-5] used a combination of the finite element method and the finite difference method to simulate wormhole generation and propagation in carbonate rocks. However, compared with Zhao’s methods, our proposed finite difference method is more easily parallelized.
Although there are a large number of numerical simulations that have been conducted to study wormhole propagation in porous media, most of them are based on the Darcy framework. While this may be adequate for scenarios where there is not a significant change in porosity, in cases where there is a large change in the porosity (and consequently the permeability), this may be quite incorrect. Clearly, there are situations in which the porosity becomes equal to unity, which implies an area of clear fluid within the porous medium. Therefore, another framework that accounts for both the porous media and clear fluid area is suggested. This is just the Darcy-Brinkman-Forchheimer (DBF) framework. In this work, the DBF framework is introduced to describe matrix acidisation for the purpose of retrieving more reasonable results. The DBF framework can predict the convective flow reasonably well in comparison with the experimental data on the condition of variable porosities [37]. Moreover, the DBF framework accounts for the boundary-layer development, macroscopic shear stress and microscopic shear stress and inertial force. A porous medium and clear fluid interface is best dealt with by the DBF framework and the continuity of velocities and stresses at the interface [38]. To the best of our knowledge, no work has ever considered simulating wormhole dynamics within the DBF framework; this work attempts to fill in this gap.

The results based on the DBF framework were compared with those based on the Darcy framework. Because much work has already focused on the factors that affect the dissolution pattern, such topics were not discussed in this work. In addition to that, to simulate the case accurately, a sufficiently fine grid with a large number of cells is needed. However, if a traditional serial code is used to simulate this large grid, it would be very inefficient. This brings up the necessity to develop a parallel code. In this work, a
parallel 2D code based on the coupled DBF and pore scale model was developed. Furthermore, the scalability of the parallel code was evaluated. The experimenting field approach to set up coefficients in the model equations was also introduced. Lastly, the details to apply the solvers to the parallel code were also given in this work. In brief, the contributions of this work can be listed as below:

1) Introduce the DBF framework and compare its results with those from the Darcy framework.
2) Develop a parallel 2D code to do simulations.
3) Introduce the experimenting field approach to set coefficients in the model equations.
4) Describe the coefficient matrix setting procedure in solver.

2. Two-Scale Model for Carbonate Dissolution

The equation system that describes wormhole propagation in porous media includes momentum balance equations (Eq. 1), mass conservation equations (Eq. 2), solute transport equations (Eq. 3) and a number of semi-empirical equations that bridge the scale differences, as described earlier (Eqs. 4-6) [30]. They can be listed as follows:

\[
\frac{\rho}{\phi} \frac{\partial u}{\partial t} + \frac{\rho}{\phi^2} \nabla \cdot uu = -\nabla p + \frac{\mu}{\phi} \nabla^2 u - \frac{\mu}{K} u - \frac{\rho F}{\sqrt{K}} u|u| u + \rho g, \tag{1}
\]

with \( F = \frac{1.75}{\sqrt{150} \phi^3} \) being the Forchheimer coefficient.

\[
\frac{\partial \phi}{\partial t} + \nabla \cdot u = 0, \tag{2}
\]

\[
\frac{\partial (\phi C_f)}{\partial t} + \nabla \cdot (u C_f) = \nabla \cdot (\phi D_e \cdot \nabla C_f) - k_c a_v (C_f - C_s), \tag{3}
\]
\[ k_c (C_f - C_s) = R(C_s), \]  
(4)

\[ \frac{\partial \phi}{\partial t} = \frac{R(C_s) a_p \alpha}{\rho_s}, \]  
(5)

\[ C_s = \frac{C_f}{1 + \frac{k_c}{k_s}}. \]  
(6)

In the above equations, \( u \) is the velocity vector, \( \rho \) is the fluid density, \( \phi \) is the porosity, \( t \) is the time and \( \mu \) is the fluid viscosity. If there is isotropic and homogeneous permeability, then \( K \) is the permeability value. \( g \) is the gravity vector, \( p \) is the pressure, \( C_f \) is the cup-mixing concentration of the acid in the fluid phase, \( D_e \) is the effective dispersion tensor, \( k_c \) is the local mass-transfer coefficient, \( a_p \) is the interfacial area available for reaction per unit volume of the medium, \( C_s \) is the concentration of the acid at the fluid-solid interface, \( R(C_s) \) is the reaction rate, \( \alpha \) is the dissolving power of the acid, \( \rho_s \) is the density of the solid phase and \( k_s \) is the surface reaction rate constant.

The equation system is solved simultaneously to obtain the pressures, the velocities and the concentrations of the acid. For this system, an update of the structural properties of the porous medium such as the porosity is needed. Eq. 1 is based on the general Navier-Stokes equation, extended by the Darcy term, the Brinkman term and the Forchheimer term. The term \(-\nabla p - \frac{\mu}{k} u + \rho g\) is the Darcy term. The term \(\frac{\mu}{\phi} \nabla^2 u\) is called the Brinkman term, which is used to account for transitional flow between boundaries.

For very high velocities in porous media, inertial effects can also become significant, so the inertial term \(\frac{\rho F}{\sqrt{K}} |u| u\), which is also called the Forchheimer term, is also added to Eq. 1. In Eq. 2, both the medium and fluid compressibility have been neglected, although they may have some effects on the wormhole generation and propagation in carbonate
rocks [10]. It is noteworthy that an additional term is added to the left hand side of Eq. 2. This term is known as the pseudo compressibility to enhance the conditioning status of the linear system. As stated in Langtangen et al. [39], in some conditions, when \( \varepsilon \) is 0, the system may be singular and cannot be solved by a linear solver. One suggested solution is to set \( \varepsilon \) to be a very small positive number to ensure an invertible coefficient matrix. Eq. 2 is therefore modified to

\[
\varepsilon \frac{\partial p}{\partial t} + \frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{u} = 0.
\]

Eq. 3 computes the concentration of the acid species. The first three terms stand for the accumulation, advection [40] and dispersion of the acid, respectively. The fourth term describes the transfer of the acid species from the fluid phase to the fluid-solid interface. Eq. 4 balances the amount of reactant transferred to the surface and the amount reacted. The mineral reactive surface area has been neglected. Eq. 5 describes the evolution of porosity in the domain as a result of reaction, and Eq. 6 shows the relationship between \( C_f \) and \( C_s \). It is noted that temperature effects [18,19] have been ignored in Eq. 1 – 5, so that only isothermal acid dissolution systems are considered in this study.

In the pore scale model, the following three equations are given:

\[
\frac{K}{K_0} = \frac{\phi}{\phi_0} \left( \frac{\phi(1-\phi_0)}{\phi_0(1-\phi)} \right)^2, \quad (7)
\]

\[
\frac{r_p}{r_0} = \sqrt{\frac{K\phi_0}{K_0\phi'}} \quad (8)
\]

\[
\frac{a_v}{a_0} = \frac{\phi r_0}{\phi_0 r_p}, \quad (9)
\]
\( r_p \) represents the average pore radius. The subscript 0 stands for the initial value. The structural properties of porous medium can be learned from the pore scale model. Eq. 7 is the Carman-Kozeny correlation, which establishes the relationship between the permeability and the porosity. Eqs. 8 and 9 show the relationship between the permeability and interfacial area.

**Figure 1** Flow chart of solution procedure.

From Eqs. 1 to 9, \( \rho, \mu, g, k_c, \alpha, \rho_s, k_s, K_0, \phi_0, r_0 \) and \( a_0 \) are considered constant, while the other variables will change in the loop of time iteration. A semi-implicit scheme
is used to solve Eqs. 1 and 2 together to derive the pressures and the velocities. The semi-implicit scheme can be written as

\[
\frac{\rho \mathbf{u}^{\ell+1} - \mathbf{u}^{\ell}}{\Delta t} = -\frac{\rho}{\phi^2} \nabla \cdot \mathbf{u}^{\ell+1} + \frac{\mu}{\phi} \nabla^2 \mathbf{u}^{\ell+1} - \frac{\mu}{K} \frac{\rho F}{\sqrt{K}} |\mathbf{u}|^{\ell+1} + \rho \mathbf{g},
\]

\[
\frac{\varepsilon p^{\ell+1} - p^{\ell}}{\Delta t} = -\nabla \cdot \mathbf{u}^{\ell+1} - \frac{R(C_s)\alpha v}{\rho_s},
\]

where \( \ell \) stands for the time step. In addition to the pressures and velocities, the values of all other variables are from time step \( \ell \), and the superscripts are abbreviated here. Then, it is suggested to use the method on Page 136-137 in Chen [41] to calculate the effective dispersion tensor \( \mathbf{D}_e \). With the new velocities and effective dispersion tensor, Eqs. 3 and 6 are ready to compute \( C_f \) and \( C_s \) using an implicit method. Furthermore, Eqs. 4, 5 and 9 can be used to derive the new porosity \( \phi \) with an implicit method. Combining the three equations, a new equation is generated as

\[
\frac{\partial \phi}{\partial t} = \frac{a_0 \alpha C_f k_c k_s (1 - \phi)}{\rho_s (k_c + k_s) (1 - \phi_0)}.
\]

With the implicit scheme, we have

\[
\frac{\phi^{\ell+1} - \phi^{\ell}}{\Delta t} = \frac{a_0 \alpha C_f^{\ell+1} k_c k_s (1 - \phi^{\ell+1})}{\rho_s (k_c + k_s) (1 - \phi_0)},
\]

\[
\phi^{\ell+1} = \frac{A + \phi^{\ell}}{1 + A},
\]

with \( A = \frac{a_0 \alpha C_f^{\ell+1} k_c k_s \Delta t}{\rho_s (k_c + k_s) (1 - \phi_0)} \). The new porosity can then be substituted into Eq. 7 to update the permeability \( K \). Finally, with Eqs. 8 and 9, the new \( \alpha_v \) can be calculated. With all the
new parameters, one can go back to Eqs. 1 and 2 to continue the loop. The solution procedure is shown in Figure 1.

3. Discretization and Experimenting Field Approach

The finite difference method is used here for discretization. After the discretization, there are two linear systems, one to solve the pressures and velocities and the other to solve the concentrations. The discretization of the first linear system will be described in the following section, and the discretization of the second linear system is similar. The experimenting field approach is adapted in this work [42-46]. While this approach has been implemented in these works for the pressure field, it is generalised to include the velocity field as well. Suppose there is a 2D domain and it is divided into 16 cells as shown in Figure 2. The velocity field is represented as arrows on the edges, and the pressure field is represented as points in the centres of the cells.

Figure 2. Discretization of the 2D domain.
In the following discussion, Eq. 1 is called the momentum equation, and Eq. 2 is called the continuity equation. The discretized equations of Eqs. 1 and 2 form a linear system. For each vertical edge of the cells, a discretized x-momentum equation can be imposed on it. Suppose the vertical edge lies in $i$-th column and $j$-th row. Eq. 1 can then be discretized as

$$a_{i,j}^{(o)} u_{x,i,j-1} + a_{i-1,j}^{(o)} u_{x,i-1,j} + a_{i,j}^{(o)} u_{x,i,j} + a_{i,j+1}^{(o)} u_{x,i,j+1} + a_{i,j+1}^{(o)} u_{x,i,j+1} +$$

$$b_{i-1,j}^{(o)} p_{i-1,j} + b_{i,j}^{(o)} p_{i,j} = c^{(o)}, \quad i = 1, ..., 5, j = 1, ..., 4, \quad (10)$$

$a$ is the coefficient of the $x$-direction velocity, $b$ is the coefficient of pressure and $c$ is a constant term on the right-hand side. The superscript $(o)$ represents the index of this equation in the linear system, which will be referenced later. The terms that have 0 subscripts are removed from the equations. In the same way, the discretized y-momentum equations are imposed on the horizontal edges. Suppose the horizontal edge lies in $i$-th column and $j$-th row. Eq. 1 can then be discretized as

$$a_{i,j}^{(o)} u_{y,i,j-1} + a_{i-1,j}^{(o)} u_{y,i-1,j} + a_{i,j}^{(o)} u_{y,i,j} + a_{i,j+1}^{(o)} u_{y,i,j+1} +$$

$$b_{i-1,j}^{(o)} p_{i-1,j} + b_{i,j}^{(o)} p_{i,j} = c^{(o)}, \quad i = 1, ..., 4, j = 1, ..., 5, \quad (11)$$

$a'$ is the coefficient of the $y$-direction velocity, $b'$ is the coefficient of the pressure and $c'$ is a constant term on the right-hand side. Similarly, the terms with 0 subscripts are removed from the equations. Under Neumann boundary condition, the $x$-momentum equations and $y$-momentum equations on the boundaries degenerate to

$$u = u_B.$$
$u_B$ is the boundary velocity. Eq. 2 is imposed in the centre of the cell so that, for the cell in $i$-th column and $j$-th row, Eq. 2 can be discretized as

$$b^{(o)}_{i,j} p_{i,j} + u_{x,i+1,j} - u_{x,i,j} + u_{y,i,j+1} - u_{y,i,j} = c^{(o)} \quad i = 1, \ldots, 4, j = 1, \ldots, 4.$$  

(12)

$b^{(o)}$ is the coefficient of pressure, and $c^{(o)}$ is the constant term on the right-hand side. It is noted that Eq. 5 is substituted into Eq. 2 to cancel the unknown $\phi$ here.

![Figure 3. Indices of unknowns in $x$ vector.](image)

By combining Eqs. 10, 11 and 12, the linear system $Ax = b$ can be generated with

$$x = (u_{x,1,1}, u_{x,2,1}, \ldots, u_{x,5,4}, u_{y,1,1}, u_{y,2,1}, \ldots, u_{y,4,5}, p_{1,1}, p_{2,1}, \ldots, p_{4,4})^T.$$  

$b$ is the right-hand-side vector and $A$ is the coefficient matrix. Figure 3 shows the indices of the unknown variables in $x$ in a more direct way. The indices of the velocity are represented by the numbers on the edges, and the indices of pressure are represented by
the numbers in the cells. Moreover, matrix \( A \) can be divided into nine sub matrices, and

\[
\mathbf{Ax} = \mathbf{b}
\]
can then be rewritten as

\[
\begin{pmatrix}
A_{xx} & A_{xy} & A_{xp} \\
A_{yx} & A_{yy} & A_{yp} \\
A_{cx} & A_{cy} & A_{cp}
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}_x \\
\mathbf{u}_y \\
\mathbf{p}
\end{pmatrix} = \mathbf{b}.
\]

\( A_{xx} \) consists of the coefficients of \( \mathbf{u}_x \) in the \( x \)-momentum equations, \( A_{xp} \) contains the coefficients of \( \mathbf{p} \) in the \( x \)-momentum equations and \( A_{xy} = 0 \) because there is no \( \mathbf{u}_y \) in the \( x \)-momentum equations. \( A_{yy} \) consists of the coefficients of \( \mathbf{u}_y \) in the \( y \)-momentum equations, \( A_{yp} \) contains the coefficients of \( \mathbf{p} \) in the \( y \)-momentum equations and \( A_{yx} = 0 \) because there is no \( \mathbf{u}_x \) in the \( y \)-momentum equations. \( A_{cx} \) consists of the coefficients of \( \mathbf{u}_x \) in the continuity equations, \( A_{cy} \) contains the coefficients of \( \mathbf{u}_y \) in the continuity equations and \( A_{cp} \) contains the coefficients of \( \mathbf{p} \) in the continuity equations.

Traditionally, an algebraic method is used to derive the equation coefficient expression by hand, which is cumbersome and error-prone. Thus, a new method called the experimenting field approach is proposed here to calculate coefficients directly without knowing their concrete expressions. It is easy to see that the numbers in Figure 3 also represent the indices of the equations in \( \mathbf{Ax} = \mathbf{b} \). The numbers on the vertical edges stand for the indices of the \( x \)-momentum equations, the numbers on the horizontal edges stand for the indices of the \( y \)-momentum equations and the numbers in the cells stand for the indices of the continuity equations. For example, the 8th equation is imposed on the edge numbered 8. It can be expressed as

\[
a_3^{(8)} u_{x3} + a_7^{(8)} u_{x7} + a_8^{(8)} u_{x8} + a_9^{(8)} u_{x9} + a_{13}^{(8)} u_{x13} + b_{46}^{(8)} p_{46} + b_{47}^{(8)} p_{47} = c^{(8)}.
\]
It is noted that the subscripts are represented by the index numbers here instead of the coordinates as before. If one sets \( u_{x8} = 1 \) and the other unknowns as 0, the left-hand-side of the equation is equal to the coefficient of \( u_{x8} \), which is \( a_{8}^{(8)} \). In the same way, the other coefficients can be calculated easily without knowing their expressions. Furthermore, it is also noted that there are five equations that contain the unknown variable \( u_{x8} \). They are the 3rd equation, the 7th equation, the 8th equation, the 9th equation and the 13th equation, as listed below

\[
a_{2}^{(3)} u_{x2} + a_{3}^{(3)} u_{x3} + a_{4}^{(3)} u_{x4} + a_{8}^{(3)} u_{x8} + b_{42}^{(3)} p_{42} + b_{43}^{(3)} p_{43} = c^{(3)},
\]
\[
a_{2}^{(7)} u_{x2} + a_{6}^{(7)} u_{x6} + a_{7}^{(7)} u_{x7} + a_{8}^{(7)} u_{x8} + a_{12}^{(7)} u_{x12} + b_{45}^{(7)} p_{45} + b_{46}^{(7)} p_{46} = c^{(7)},
\]
\[
a_{3}^{(8)} u_{x3} + a_{7}^{(8)} u_{x7} + a_{8}^{(8)} u_{x8} + a_{9}^{(8)} u_{x9} + a_{13}^{(8)} u_{x13} + b_{46}^{(8)} p_{46} + b_{47}^{(8)} p_{47} = c^{(8)},
\]
\[
a_{4}^{(9)} u_{x4} + a_{8}^{(9)} u_{x8} + a_{9}^{(9)} u_{x9} + a_{10}^{(9)} u_{x10} + a_{14}^{(9)} u_{x14} + b_{47}^{(9)} p_{47} + b_{48}^{(9)} p_{48} = c^{(9)},
\]
\[
a_{8}^{(13)} u_{x8} + a_{12}^{(13)} u_{x12} + a_{13}^{(13)} u_{x13} + a_{14}^{(13)} u_{x14} + a_{18}^{(13)} u_{x18} + b_{50}^{(13)} p_{50} + b_{51}^{(13)} p_{51} = c^{(13)}.
\]

If one sets \( u_{x8} = 1 \) and the other unknowns as 0, one can derive five coefficients at the same time: \( a_{8}^{(3)}, a_{8}^{(7)}, a_{8}^{(8)}, a_{8}^{(9)} \) and \( a_{8}^{(13)} \). Thus, this method needs only \( N \) loops to derive all the coefficient values of the linear system, with \( N \) equalling the number of unknowns. It is emphasised that the method can calculate coefficient values automatically, and there is no need to know the details of their algebraic expressions.
Figure 4. $x$-direction velocity field in nine shifts. The bold arrows stand for 1 values, and the other arrows stand for 0 values.

The number of loops can be reduced even further. All the unknowns are divided into three fields: the $x$-direction velocity field, the $y$-direction velocity field and the pressure field. The $x$-direction velocity field corresponding to the grid in Figure 3 is shown in Figure 4. In the $x$-momentum equations, if one sets $u_{x,1,1} = 1$, one can derive coefficient values from the 1st, 2nd and 6th equations and cannot derive any coefficient values from the other equations because the other equations do not contain the variable $u_{x,1,1}$. With this philosophy, when one sets $u_{x,1,1} = 1$, one can set $u_{x,4,1} = 1$, $u_{x,1,4} = 1$ and $u_{x,4,4} =$
1 at the same time, which is shown as the bold arrows in the first plot of Figure 4. In fact, one can set \( u_x \) every other two as 1 in each direction at the same time. Then, one can shift the bold arrows by one place and set another four \( u_x \) as 1, which is shown in the second plot of Figure 4. The figure shows the entire shift procedure. Thus, a total of only nine shifts are needed to derive all the coefficient values of \( u_x \) in the \( x\)-momentum equations, which is \( A_{xx} \). The loop number of nine is much less than the loop number of 20 before the improvement. Before the acceleration, \( O(N) \) loops are needed to derive all the coefficients, but now, only nine shifts are needed, which is \( O(1) \). When \( N \) is very large, which is very common in large-scale problems, the acceleration is tremendous.

To derive the coefficient values of the pressures in the \( x\)-momentum equations, the pressure field is shifted. Because each pressure variable appears only in the two equations of its left and right side, one can set every other one as 1 in the \( x\)-direction. A total of only two shifts are needed to derive all the \( p \) coefficient values in the \( x\)-momentum equations. All the values make up the nonzero entries of the matrix \( A_{xp} \). The shift procedure is shown in Figure 5. The same philosophy is held to derive the coefficient values of the \( y\)-momentum equations, which are also the nonzero entries of matrices \( A_{yy} \) and \( A_{yp} \). The field shift procedure of the continuity equations is similar, and the description is omitted here.
Figure 5. Pressure field in two shifts. The black points stand for 1 values, and the white points stand for 0 values.

In Eq. 1, the porosities \( \phi \) and the permeability tensors \( k \) on the edges are calculated using the harmonic weighting method. Eq. 3 is imposed on the centre of the cells, and the concentrations of acid \( C_f \) on the edges are calculated by the upwind method.

4. Parallelization

For the wormhole issue, it is necessary to have a very fine mesh to describe the natural distribution of the porosities in the porous medium, which requires a huge number of grid cells. Because the grid size becomes large, it is reasonable to leverage parallelism to speed up the simulation procedure. To the best of our knowledge, no work has ever tried to use parallelism to simulate wormhole propagation before. Thus, in this part, the parallel method is used to accelerate wormhole simulations. Suppose there are four processors, and then the domain in Figure 2 can be decomposed as shown in Figure 6. It has been already stipulated in the former section that an edge of a cell can represent either a velocity on the edge or a momentum equation imposed on the edge. Similarly, the
central point of a cell can represent either a pressure in the cell or a continuity equation imposed on the point. With this philosophy, the notations in Figure 6 are easy to understand. All the variables and equations represented by red, black, green and blue edges and points are allocated to Processors 0, 1, 2 and 3, respectively. In this manner, no two processors have overlapping variables and equations. Such a decomposition strategy enables the achievement of load balance for each processor. Moreover, it reduces the communication cost among the processors to the lowest level.

**Figure 6.** Domain decomposition strategy. The red, black, green and blue parts are allocated to Processors 0, 1, 2 and 3, respectively.
Figure 7. Overlapping edges needed by $x$-momentum equations (1), $y$-momentum equations (2) and continuity equations (3) in Processor 0. All variables and equations in Processor 0 can be shown in (4).

However, it is still necessary to allow some overlapping data in one processor. Take Processor 0 as an example. In the first plot of Figure 7, Numbers 1 to 12 mark all the variables and equations in Processor 0. At first, it is verified what overlapping data is needed to generate the $x$-momentum equations in Processor 0. For example, to derive the coefficient of $u_{x,13}$ in the 2nd equation (the $x$-momentum equation imposed on the edge numbered 2), one has to set $u_{x,13} = 1$. However, $u_{x,13}$ is not in Processor 0 but rather is in Processor 1. Thus, Processor 0 has to store the overlapping variable $u_{x,13}$ and the overlapping equation imposed on Edge 13. The other overlapping edges of such type in
Processor 0 can be derived in the same way, which are shown as the yellow edges in the first plot of Figure 7. It is also noted that the 13th equation needs the velocities \( u_{y,6} \), \( u_{y,8} \), \( u_{y,19} \) and \( u_{y,21} \) to calculate its advection term, so that Edges 19 and 21 are included in Processor 0. Such a type of overlapping edges in Processor 0 is represented by the grey edges in the first plot of Figure 7. With the same philosophy, all the overlapping edges needed by the \( y \)-momentum equations and continuity equations in Processor 0 are shown in the second and third plots of Figure 7 respectively. Finally, by combining all the overlapping variables and equations in Figure 7 (1), (2) and (3) together, all the variables and equations in Processor 0 can be observed in Figure 7 (4).

In addition to creating the overlapping edges, Processor 0 also needs to communicate with the neighbouring processors to derive the necessary variable values. For example, in the first plot of Figure 7, all the \( y \)-direction velocities on the grey edges should be acquired from the neighbouring processors. In the second plot of Figure 7, all the \( x \)-direction velocities on the grey edges should be transferred to Processor 0 from the neighbouring processors. There is no need to transfer pressures among the processors. All the communications above constitute the communication cost of the parallel code. The overlapping edges and the communication cost of the other processors hold to the same philosophy as Processor 0, so the description is omitted here.
In the parallel code, the structure of the linear system $Ax = b$ is different from its counterpart in the serial code. The library Hypre [47] is used to solve the system. In Hypre, the unknown variables in vector $x$ are arranged processor by processor, which means that the variables in Processor 0 are put into $x$ at first, and then it is the variables in Processor 1 that are put into $x$ amongst others. In each processor, the variables are still arranged in the order of the $x$-direction velocities, then $y$-direction velocities and last pressures. According to the domain decomposition strategy in Figure 6, the indices of the unknown variables in $x$ in the parallel code are shown in Figure 8. It can be easily observed that the indices in Figure 8 also represent the indices of equations in the linear system. Because the indices of the unknowns in $x$ and the indices of the equations have changed, the locations of the nonzero entries of matrix $A$ also need to be changed. The nine sub matrices of $A$ in the serial code no longer exist.
The linear system $Ax = b$ in Hypre can be observed in Figure 9. P0 to P3 represent Processors 0 to 3, respectively. From the figure, it can be observed that each processor occupies one part of the data of the linear system, which is separated by the three dash lines. After the processors generate their own data, the solver in Hypre assembles them together and solves the linear system.

5. Validation

A parallel code with FORTRAN 90 and MPI was developed to simulate the wormhole forming procedure. The preconditioner DS [48] and the solver GMRES in Hypre are chosen to solve $Ax = b$. To validate the correctness of the code, the benchmark problem shear-driven cavity flow [49] was reproduced with it on the supercomputer Shaheen [50]. In the problem, there is a laminar incompressible flow.
inside a square cavity whose top wall moves with a uniform velocity, as is shown in Figure 10. The length of the square cavity is 1 m, and the wall velocity is 1 m/s. The Reynolds number \( (Re) \) is 100. The gravity effect is ignored. The grid has \( 100^2 \) cells. Since \( \varepsilon \) is zero, the iterative solver cannot solve the linear system, so the direct sparse solver UMFPACK [52] is used. When the flow becomes stable, the simulation results are shown from Figure 11 to Figure 16. Figure 11 to Figure 13 can be compared with the results from the commercial software such as CFD [51]. Figure 14 shows the streamlines in the cavity, which can be compared with Figure 3 in [49]. Their configurations are very similar. Figure 15 shows the \( y \)-direction velocity component along the horizontal centreline, and Figure 16 shows the \( x \)-direction velocity component along the vertical centreline. Both of the present results were compared with Ghia’s data in [49]. It is found that the present results match Ghia’s data to eyeball resolution. All the above experimental results validate the correctness of the code.

![Figure 10](image)

**Figure 10.** Shear-driven fluid flow in a square cavity. \( u \) stands for the \( x \)-direction boundary velocity, and \( v \) stands for the \( y \)-direction boundary velocity.
Figure 11. Pressure field of the 2D domain.

Figure 12. $x$-direction velocity field in the 2D domain.
Figure 13. $y$-direction velocity field in the 2D domain.

Figure 14. Streamlines of the 2D domain.
6. Sensitivity Analysis
A sensitivity analysis has to be carried out to make sure that the simulation grid is fine enough. A small porous medium of 0.2-m by 0.2-m was analysed based on the consideration of saving computing time. Three types of grid with $20^2$, $40^2$ and $60^2$ cells were tested. The initial porosity in the porous medium is 0.2 everywhere, and the initial permeability of the porous medium is $10^{-8}$ m$^2$ everywhere. An acid flow is injected from the left boundary of the porous medium at a velocity of 0.01 m/s. A constant pressure of $10^5$ Pa is imposed on the right boundary of the porous medium. The upper and lower boundaries of the porous medium are closed. The initial pressure inside the porous medium is also $10^5$ Pa. Two high-porosity entries are set on the left boundary of the porous medium, one 0.05 metre above the bottom with a porosity of 0.6 and the other 0.1 metre above the bottom with a porosity of 0.5. The simulation period is $10^4$ s, and each time step is $10^2$ s. The simulation is based on the DBF framework. The other properties of the acid flow and the porous medium are shown in Table 1. The parameter $\varepsilon$ is set to 0. In the experiment, the condition number of the generated linear system is very large, and popular preconditioners and solvers in Hypre cannot achieve convergence in a reasonable number of iterations, so the sparse direct solver UMFPACK is used here. The change in the average left boundary pressure is shown in Figure 17. The change in the average permeability is shown in Figure 18. The porosity fields in the porous medium at the end of the simulation are shown in Figure 19.

From Figure 17, it can be observed that the three curves generated by the three resolutions are almost the same. In Figure 18, the curve generated by the grid of $20^2$ cells is slightly different from the other two curves, which are very close to each other. From Figure 19, it also can be observed that the two porosity fields of (b) and (c) are very
similar to each other. From the observations, we conclude that the difference between the results from $40^2$ and $60^2$ cells is negligible, so the grid of $40^2$ cells is fine enough to simulate the 0.2-meter by 0.2-meter porous medium. In such a resolution, the length of a cell is 5 mm.

Table 1. Properties of acid flow and porous medium

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viscosity of acid flow ($\mu$)</td>
<td>$10^{-2}$ Pa·s</td>
</tr>
<tr>
<td>Density of acid flow ($\rho$)</td>
<td>$10^3$ kg/m$^2$</td>
</tr>
<tr>
<td>Longitudinal dispersion coefficient</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>Transverse dispersion coefficient</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Molecular diffusion coefficient</td>
<td>$10^{-9}$</td>
</tr>
<tr>
<td>Surface reaction rate constant ($k_s$)</td>
<td>10 m/s</td>
</tr>
<tr>
<td>Local mass-transfer coefficient ($k_c$)</td>
<td>1 m/s</td>
</tr>
<tr>
<td>Dissolving power of acid ($\alpha$)</td>
<td>0.1 kg/mol</td>
</tr>
<tr>
<td>Density of the rock ($\rho_s$)</td>
<td>2500 kg/m$^2$</td>
</tr>
<tr>
<td>Initial interfacial area ($a_0$)</td>
<td>0.5 m$^{-1}$</td>
</tr>
<tr>
<td>Initial concentration of acid flow ($C^0_f$)</td>
<td>10 mol/m$^2$</td>
</tr>
</tbody>
</table>

Figure 17. Change of average left boundary pressure in the three resolutions.
Figure 18. Change of average permeability in the three resolutions.

Figure 19. Porosity field at the end of simulation in the three resolutions: (a) $20^2$, (b) $40^2$ and (c) $60^2$. 
7. Dissolution Experiment

After the sensitivity analysis, the development of wormholes in a 1-m by 0.2-m porous medium using both the Darcy and DBF frameworks can be simulated. The initial porosities in the porous medium comply with the uniform distribution, and the range is from 0.05 to 0.35. The initial permeability of the porous medium also obeys the uniform distribution, and the average value is $10^{-8}$ m$^2$. The other physical parameters are the same as the sensitivity analysis. The simulation period is $10^5$ s, and each time step is 1000 s.

From the conclusion of the sensitivity analysis, the simulation grid should have 200 cells in the length direction and 40 cells in the width direction. The distributions of the porosities in both frameworks at five different moments were compared, as is shown in Figure 20. The development of four properties in both frameworks was also compared, including the average porosity, the average permeability, the average interfacial area and the flux of the right boundary in the porous medium. Figure 21 to Figure 24 show the differences in the four properties in both frameworks during the procedure of matrix acidisation.

From Figure 20, it can be observed that at the beginning of the development, both frameworks show the formation of some small fingers. As time elapses, two major fingers (upper finger and lower finger) appear. However, the lower finger develops much faster than the upper finger. In the Darcy framework, the upper finger even disappears at the $10^5$th second. Furthermore, it can be observed that there are two major differences between the simulation results of both frameworks. One difference is that the major finger in the Darcy framework is shorter than the one in the DBF framework at all five moments. For example, at the $10^5$th second, the major finger of the DBF framework has
achieved a breakthrough, while the tip of major finger of the Darcy framework is still near the right boundary of porous medium. The other difference is that the major finger in the DBF framework is thinner than the one in DBF framework at all five moments. Both of these differences show that the DBF framework more accurately simulated the wormhole phenomenon than the Darcy framework.

**Figure 20.** Distributions of porosities in the Darcy framework (left) and DBF framework (right) at five different moments. The range of colour from blue to red indicates the range of porosity from 0 to 1.
From Figure 21, it may be observed that the average porosities in both frameworks are always increasing, which indicates that the matrix is eaten by the acid. The DBF and Darcy framework curves are almost the same before the $3.5 \times 10^4$th second, but Darcy framework curve is higher than the DBF framework curve after that moment. The DBF framework curve tends to converge to a limit, while the Darcy framework curve tends to increase faster. Thus, the DBF framework curve represents the wormhole phenomenon better than Darcy framework curve. Figure 22, shows the increase of the average permeability in both frameworks, with the DBF framework curve increasing much faster than the Darcy framework curve after the $8.9 \times 10^4$th second. In Figure 23, both curves are decreasing over the whole simulation procedure, which indicates the decreasing of the average interfacial area with the formation of wormholes. However, the average interfacial area of the Darcy framework decreases faster than that of the DBF framework after the $3.5 \times 10^4$th second. The moment coincides with the moment in Figure 21 when the faster increase of Darcy framework curve can be observed. In Figure 24, the right boundary flux of the DBF framework is stable over most of the simulation time at a value of approximately 0.01 m/s, which is the injection velocity of the acid flow. However, many oscillations can be observed in the Darcy framework curve. This also demonstrates that the numerical results of the DBF framework are more stable than those of the Darcy framework.
Figure 21. Average porosity curves in the Darcy and DBF frameworks.

Figure 22. Average permeability curves in the Darcy and DBF frameworks.
Figure 23. Average interfacial area curves in the Darcy and DBF frameworks.

Figure 24. Right boundary flux curves in the Darcy and DBF frameworks.
8. Performance Evaluation

To evaluate the parallel performance of the code, the benchmark shear driven cavity flow problem was tested on Shaheen again. The simulation time is 60 s, and there are 1000 time steps. The concentration field is constant in this case, so there is no need to solve the linear system in the concentration field. In this experiment, only one linear system is solved for the pressure and velocity fields. The grid has $100^2$ cells. All the I/O cost is ignored. In order to solve the linear system using Hypre, $\varepsilon$ is set to 0.1. The average total run time and solver time for different number of processors can be observed in Table 2 and Figure 25. From this data, it can be observed that the speedup of the solver is very close to the ideal linear speedup. In addition to that, the whole parallel code achieves a superlinear speedup. As noted by Wilkinson [53], one reason for this superlinear speedup is the efficient use of RAM. The working set of each processor decreases with the increase in the number of processors. A smaller working set means fewer or no cache, which contributes to the appearance of a superlinear speedup. The other reason is the decrease in the number of disk I/O for each processor with the increase in the number of processors.

Table 2 Average run time and solver time of the code when solving the shear driven cavity flow problem. P is the number of processors, and the unit is second.

<table>
<thead>
<tr>
<th>P</th>
<th>Run Time</th>
<th>Solver Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33512.1</td>
<td>5478.5</td>
</tr>
<tr>
<td>2</td>
<td>10130.1</td>
<td>2810.9</td>
</tr>
<tr>
<td>4</td>
<td>3138.9</td>
<td>1470.0</td>
</tr>
<tr>
<td>8</td>
<td>1162.2</td>
<td>747.6</td>
</tr>
</tbody>
</table>
9. Conclusions and Future Work

In this work, 2D wormholes were simulated with the Darcy and DBF frameworks. By comparing the results from the two frameworks, we conclude that the wormhole configuration simulated by the DBF framework is more accurate than that of the Darcy framework. Furthermore, the numerical results of the DBF framework are more stable than those of the Darcy framework. Finally, the parallel code achieved superlinear scalability when tested on Shaheen.

In the future, more properties simulated by the two frameworks such as the optimal injection rate and the breakthrough time will be compared. The study of wormhole properties in the 3D case will also be investigated.
References


[51] http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics