Review Article

Unsplit Youngs Method for Tracking the Moving Interface of Expansible Grout

Xiaolong Li ¹, **Tiantian Zhang** ¹, **Meimei Hao** ^{1,2}, **Bei Zhang** ¹, **Wenyi Yao** ³,
and **Zhenzhou Shen** ³

¹School of Water Conservancy Engineering, Zhengzhou University, Science Road, No. 100, Zhengzhou 450001, China

²School of Civil Engineering, Zhengzhou University, Science Road, No. 100, Zhengzhou 450001, China

³Yellow River Institute of Hydraulic Research, Shunhe Road, No. 45, Zhengzhou 450003, China

Correspondence should be addressed to Meimei Hao; haomm1213@outlook.com, Bei Zhang; beizhang@126.com, and Wenyi Yao; wyyao@yrihr.com.cn

Received 8 January 2020; Revised 10 October 2020; Accepted 15 October 2020; Published 29 October 2020

Academic Editor: Michele Scarpiniti

Copyright © 2020 Xiaolong Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurately tracking the moving interface is one of the most critical challenges in numerically modeling grout diffusion in fractures. The traditional Youngs method does not consider fluid transport to the oblique adjacent cell, which leads to repeated calculation and error accumulation. Based on the interface reconstruction technology, this paper introduced the idea of the unsplit algorithm to capture the grout-air interface. The unsplit interface advancement formulas were derived considering the fluid fluxes from the central cell to the oblique adjacent cells in the structured cell system. The numerical method based on the unsplit algorithm was validated by examples of predicting the propagation of expansible grout in a fracture. Results show that the diffusion pattern and grout front position calculated by the proposed method are in better agreement with the analytical solution compared with the traditional split algorithm. The new method can effectively improve the grout-air interface tracking accuracy.

1. Introduction

Grouting has been widely used to reinforce and control the seepage of rock mass in the field of infrastructure construction, such as traffic and water conservancy [1–3]. With the development of modern macromolecular science, polymer grouting material with the characteristic of self-expansion and its high-pressure injection technology have more extensive application in recent years [4–6]. When the two-component polymer grout is injected into the gap or fracture, it rapidly expands and solidifies due to the chemical reaction. The excellent properties can be utilized to fill the gap, compact the soil, strengthen the rock mass, and seal the flow channels.

The diffusion of polymer grout mainly depends on the secondary pressurization caused by self-expansion force, which is produced by the chemical reaction. This is different from the nonexpansive grout which is driven by the grouting pressure [7, 8]. The expansion property of the polymer is

influenced by environmental pressure, the extent of reaction, and other factors. Therefore, the grouting mechanism of polymer grout is more complex compared with the traditional nonexpansive grout. Although great progress has been made in grouting equipment, technology, and engineering practices, the study on the grouting mechanism still lags significantly behind it due to the concealment of grouting construction and the particularity of polymer material.

Numerical simulation is an important way to study the diffusion mechanism of grout in rock fracture. Hassler et al. established a two-dimensional fracture network model to predict the grout diffusion process [9]. Eriksson et al. further studied the impact of filtration and fracture aperture on the diffusion behavior of Bingham grout in the fracture network [10]. Based on the Monte Carlo method, the two-dimensional or three-dimensional discrete fracture network was generated randomly to investigate the grouting mechanism [11]. The abovementioned studies construct fracture networks in a deterministic or random

way based on the one-dimensional channel model, and the grout flow is assumed as a monophasic flow. However, the grout may not be a one-dimensional flow in a large-size fracture. Therefore, the replacement between grout and air (or water), which reflects the movement of the grout front, should be paid attention.

In recent years, considering the grout diffusion as multiphase flow, the study on the refined simulation of grout diffusion behavior has attracted much attention. Liu et al. simulated the change of grout front with time and pressure distribution in a fracture by COMSOL software [12]. Similarly, Zhao [13] explored the diffusion performance of cement grout in the three-dimensional cross fracture. Accurate interface tracking is crucial for the prediction of the grout front. Li [14, 15] simulated expansible grout spread in a two-dimensional flow field based on the split Youngs algorithm. However, the split algorithm only calculated the fluid fluxes from the central cell to the top, bottom, left, and right cells, which do not fit the actual fluid flow.

In order to capture the moving interface more accurately, the unsplit algorithm was developed [16, 17]. Depending on the adopted coordinate system and unit fluid form, the pattern and ways of implementation can be different. The new algorithm considers the additional fluid transport to the oblique adjacent cell and avoids the problem of the repeated calculation of the fluid flux in the traditional Youngs method.

In this paper, an unsplit Youngs method is used to track the moving interface of expansible polymer grouting a fracture. Assuming that the unit fluid is a trapezoid in the cell containing the interface, the expressions of fluid flux from the central cell to the oblique adjacent cells are derived under the structured cell system. Then, the interface between grout and air is advanced based on the fractional volume. The diffusion process of expansible grout in a fracture is studied by the unsplit algorithm. Compared with the analytical solution and the traditional split Youngs algorithm, the unsplit Youngs method significantly improves the accuracy of tracking the moving interface.

2. Volume of Fluid Method

Tracking the moving interface is required when the flow of two-phase or multiphase fluid is modeled. Currently, the interface tracking method includes the marker and cell (MAC) method, Level-Set method, and Volume of Fluid (VOF) method [18–20], among which the VOF method is the most widely used. It can guarantee the sharp, non-oscillatory interface that is desirable in immiscible multifluid flow simulations and essential in free-surface problems on stationary meshes. The fractional volume of one fluid in each cell is expressed by the function of the volume of fluid. The interface is reconstructed by the change of the volume of fluid with time; thus, the position, shape, and change of the moving interface can be tracked.

In a two-dimensional system, the calculation area is assumed as Ω . The area where fluid A locates is Ω_1 , and the area where fluid B locates is Ω_2 . A scalar function is defined as follows:

$$\lambda(x, y, t) = \begin{cases} 1, & (x, y) \in \Omega_1, \\ 0, & (x, y) \in \Omega_2. \end{cases} \quad (1)$$

When the flow field consists of two immiscible fluids, the scalar function $\lambda(x, y, t)$ follows continuity equation:

$$\frac{\partial \lambda}{\partial t} + u \frac{\partial \lambda}{\partial x} + v \frac{\partial \lambda}{\partial y} = 0, \quad (2)$$

where t is time, u is the velocity in X , and v is the velocity in Y .

The calculation area is divided into multiple non-coincident cells, ΔV_{ij} is the volume of a cell, and the integral of $\lambda(x, y, t)$ on one cell can be written as follows:

$$F_{ij} = \frac{1}{\Delta V_{ij}} \int_{I_{ij}} \lambda(x, y, t) dV. \quad (3)$$

F is the fractional fluid volume and follows the continuity equation:

$$\frac{\partial F}{\partial t} + u \frac{\partial F}{\partial x} + v \frac{\partial F}{\partial y} = 0. \quad (4)$$

The above equation is the VOF equation. The volume fraction F is solved by equation (4). There are three cases: when $F = 1$, the fluid occupies the entire cell; when $F = 0$, the cell is empty; and when $0 < F < 1$, there is two-phase fluid in the cell. Based on the value of F , the corresponding interface can be constructed.

3. Youngs Method

The key to tracking the moving interface is to solve the VOF equation [21]. There are many methods at present, such as the Hirt–Nichols method, the Flair method, the Youngs method, and the FCT method. However, the Youngs method based on geometric principle has higher accuracy compared with other methods [19, 20].

The basic implementation steps of the Youngs method include

- (i) Constructing the fluid interface according to the fractional volume in the cell
- (ii) Advancing the moving interface by updating the VOF function according to the current flow velocity
- (iii) Repeatedly executing the above two steps in each time step.

3.1. Construction of the Fluid Moving Interface. The Youngs method constructs the approximate interface with a sloped line segment in a cell base on the geometric principle. As mentioned above, there are three cases in the calculated cell, namely,

- (1) The cell is full of fluid A, $F = 1$
- (2) The cell is an interface unit, $0 < F < 1$
- (3) The cell is filled with the fluid B, $F = 0$

Case 1 and Case 3 do not reconstruct the interface. For Case 2, the normal vector of the interface in the cell is calculated firstly, as shown in

$$\left. \begin{aligned} n_{ij}^x &= \frac{(F_{i+1,j+1} + 2F_{i+1,j} + F_{i+1,j-1} - F_{i-1,j+1} - 2F_{i-1,j} - F_{i-1,j-1})}{\delta x} \\ n_{ij}^y &= \frac{(F_{i+1,j+1} + 2F_{i,j+1} + F_{i-1,j+1} - F_{i+1,j-1} - 2F_{i,j-1} - F_{i-1,j-1})}{\delta y} \end{aligned} \right\} \quad (5)$$

Then, the angle β that the interface makes with the x -axis is determined according to the normal vector:

$$\tan \beta = \frac{-n^x}{n^y}. \quad (6)$$

The angle β is normalized to ω :

$$\tan \omega = \tan \beta \left(\frac{\delta x}{\delta y} \right), \quad (7)$$

where δx is the length of a fluid cell in the x -direction and δy is the length of the cell in the y -direction, as shown in Figure 1.

According to the angle ω and VOF function in the cell, there are 16 types of interface combinations in the cell, which can be simplified into four cases by symmetry and inversion. The four types of interfaces are shown in Figure 2. The discriminant method is as follows.

if $\omega < \pi/4$

if $F \leq 0.5 \tan \omega$

case I

else if $F \leq 1.0 - 0.5 \tan \omega$

case II

else

case IV

else

if $F \leq 0.5 \cot \omega$

case I

else if $F \leq 1.0 - 0.5 \cot \omega$

case III

else

case IV

Once the case has been determined, the four side fractions of the cell, which is the fraction of the top, bottom, left, and right sides of the cell that lie within the fluid (s_t, s_b, s_l, s_r) can be calculated according to the angle ω and the VOF function. The subscripts t, b, l , and r represent up, down, left, and right, respectively, as shown in Figure 2(d) below.

Taking case III as an example, the side fractions of the cell can be calculated as follows:

$$\left. \begin{aligned} s_t &= F - 0.5 \cot \omega \\ s_r &= 1 \\ s_b &= F + 0.5 \tan \omega \\ s_l &= 0 \end{aligned} \right\} \quad (8)$$

After obtaining the side fractions of each cell, the straight-line segment that represents the fluid interface in the cell can be plotted.

3.2. Advancement of the Moving Interface (Update VOF Function). According to the current velocity, the fluxes through sides of the cell to adjacent cells in a time step are calculated. The VOF function of each cell is updated by accumulating the inflow and the outflow. The moving interface is determined and advances according to the updated VOF function.

The Youngs method was proposed in 1982, but very few details of flux calculation were given in the original paper [22]. For this reason, Rudman [20] deduced the relevant formulas by the direction split method. From then on, the Youngs method began to draw attention and be used widely.

3.2.1. Split Youngs Algorithm

(1) Calculate Fluxes. Considering the horizontal and vertical flow of the fluid through the interface of the cell, the fluxes from the central cell to the adjacent ones, including the top, bottom, left, and right, are calculated based on the split Youngs algorithm, represented by f_t, f_b, f_l , and f_r , respectively, as shown in Figure 3.

Taking case III (in Figure 2) as an example, according to the geometric law, the fluxes can be calculated as follows.

(1) Top face:

if $u_t > 0$

$$f_t = u_t \delta t (s_t \delta_x + 0.5 u_t \delta_x \cot \beta)$$

(2) Bottom face:

if $u_b < 0$

$$f_b = \text{abs}(u_b) \delta t (s_b \delta_x - 0.5 \text{abs}(u_b) \delta t \cot \beta)$$

(3) Left side:

if $u_l < 0$

$$\text{if } \text{abs}(u_l) \delta t \leq (1 - s_b) \delta_x$$

$$f_l = 0$$

$$\text{elseif } \text{abs}(u_l) \delta t \leq (1 - s_t) \delta_x$$

$$f_l = 0.5 [\text{abs}(u_l) \delta t - (1 - s_b) \delta_x]^2 \tan \beta$$

else

$$f_l = \text{abs}(u_l) \delta t \delta y - (1 - c) \delta x \delta y$$

(4) Right side:

if $u_r > 0$

$$\text{if } u_r \delta t \leq s_t \delta_x$$

$$f_r = u_r \delta t \delta y$$

$$\text{elseif } u_r \delta t \leq s_b \delta_x$$

$$f_r = u_r \delta t \delta y - 0.5 (u_r \delta t - s_t \delta_x)^2 \tan \beta$$

else

$$f_r = F \delta x \delta y$$

where u_t, u_b, u_r , and u_l represent the interface velocity in the top, bottom, left, and right, respectively. t is the time step. The fluxes of cases I, II, and IV in four directions can be calculated in the same way.

FIGURE 1: Schematic of the fluid cell.

FIGURE 2: Four types of interface.

FIGURE 3: Schematic of fluid transport calculation for split Youngs algorithm.

(2) *Update VOF Function.* The fractional volume in the cell can be updated by calculating the fluxes from the central cell to adjacent ones during the time δt .

Given the volume fraction of the fluid in the cell (i, j) is F^t at time t , after the time δt , the increments of fractional volume in the cells $(i, j+1)$, $(i, j-1)$, $(i-1, j)$, $(i+1, j)$ caused by the fluid motion are f_t , f_b , f_l and f_r , respectively. The volume fraction in the cell at time $t + \delta t$ is

$$F^{t+\delta t}(i, j) = F^t(i, j) - \frac{(f_t + f_b + f_l + f_r)}{(\delta x \delta y)}. \quad (9)$$

The fractions of fluid volume in the four adjacent cells at time $t + \delta t$ are, respectively, corrected as follows:

$$\left. \begin{aligned} F^{t+\delta t}(i, j+1) &= F^t(i, j+1) + \frac{f_t}{(\delta x \delta y)} \\ F^{t+\delta t}(i, j-1) &= F^t(i, j-1) + \frac{f_b}{(\delta x \delta y)} \\ F^{t+\delta t}(i+1, j) &= F^t(i+1, j) + \frac{f_r}{(\delta x \delta y)} \\ F^{t+\delta t}(i-1, j) &= F^t(i-1, j) + \frac{f_l}{(\delta x \delta y)} \end{aligned} \right\}. \quad (10)$$

Conducting the same calculation for each cell can update the fractional volume of the flow field.

The computing process of fluid fluxes shows that the split Youngs algorithm just includes fluxes to four adjacent cells (top, bottom, left, and right) from the central cell. The fluid flow towards the oblique cell is ignored, which may lead to some deviations from the real.

3.2.2. Unsplit Youngs Algorithm. In order to improve tracking accuracy, an unsplit operator algorithm was introduced to advance the interface. The method not only calculates the fluxes from the central cell to the top, bottom, left, and right cells but also considers the transport to four oblique direction cells including upper right, upper left, lower left, and lower right, as shown in Figure 4. The fluxes in eight directions are denoted as C_t , C_b , C_l , C_r , C_{cor1} , C_{cor2} , C_{cor3} , C_{cor4} , and the subscripts cor1, cor2, cor3, and cor4 represent the upper right corner, the upper left corner, the lower left corner, and the lower right corner, respectively.

Hu et al. [16] assumed the unit fluid form is a triangle in the cell containing the interface. Five cases were listed when $U_t > 0$ and $U_r > 0$. The relationship between the above eight fluxes was described and f_t , f_b , f_l and f_r were calculated. However, the analysis of the pattern of fluid flow is less comprehensive, and the flux calculation for different cases has not been explained in detail. Liu et al. [17] studied the unsplit transport method in the Eulerian method. The expressions of fluid flux were given in the cylindrical coordinate system. We further analyze the unsplit transport pattern and derive the formulas of interface advancement based on trapezoid unit fluid (see Figure 2(c)) in rectangular coordinates. The implementation is shown as follows.

(1) *Fluid Fluxes in the Orthogonal Directions.* The fluxes from the central cell to four adjacent cells were supposed as C_t , C_b , C_l and C_r . Figure 5 shows the pattern of fluid transport from the central cell (Figure 5(a)) to the upper right cell (Figure 5(b)), the upper left cell (Figure 5(c)), the lower left cell (Figure 5(d)), and the lower right cell (Figure 5(e)) from t^n to t^{n+1} .

According to Figures 5(b)–5(e), the equations can be expressed as follows:

$$\begin{aligned} C_t &= f_t - C_{cor1}, \\ C_r &= f_r - C_{cor1}, \end{aligned} \quad (11)$$

$$\begin{aligned} C_t &= f_t - C_{cor2}, \\ C_l &= f_l - C_{cor2}, \end{aligned} \quad (12)$$

$$\begin{aligned} C_l &= f_l - C_{cor3}, \\ C_b &= f_b - C_{cor3}, \end{aligned} \quad (13)$$

$$\begin{aligned} C_r &= f_r - C_{cor4}, \\ C_b &= f_b - C_{cor4}. \end{aligned} \quad (14)$$

It indicates that the fluxes in four oblique directions need to be calculated first.

(2) *Fluid Fluxes in the Oblique Directions.* Taking the case of fluid flow from the central cell to the upper right cell (Figure 5(b)), the oblique flux C_{cor1} was derived. The fluid could be in nine locations after the oblique movement, as shown in Figure 6. “+” indicates the grid line, the dotted polygon is the position before the fluid moves, solid polygon indicates the current position, and the shaded portion shows

FIGURE 4: Schematic of fluid transport calculation for unsplit Youngs algorithm.

the fluid flowing into the upper right cell. The shaded area is the required flux C_{cor1} .

In Figure 7, l_m is the height of the polygon fluid beyond the intersected horizontal cell line in case III, l_b is the length of the horizontal cell line intersected with the fluid polygon, and θ is the angle of the fluid polygon bevel inclination.

C_{cor1} is calculated as follows:

if $f_t > 0$ and $f_r > 0$ then

(excluding Figure 6 (1), (3), (6), (8), (9))

$$l_m = u_t \delta t$$

$$l_b = l_m \cot \theta + s_t \delta x$$

if $l_b > u_r \delta t$ (two cases as Figure 6 (2), (5))

$$C_{cor1} = f_r - u_r \delta t (\delta y - u_t \delta t)$$

else (two cases as Figure 6 (4), (7))

$$C_{cor1} = f_t$$

In the same way, the flux expressions of case III during the time δt from the central cell to the upper left (Figure 5(c)), lower left (Figure 5(d)), and lower-right corner (Figure 5(e)), C_{cor2} , C_{cor3} , and C_{cor4} , can be written as follows.

For C_{cor2} ,

if $f_t > 0$ and $f_l > 0$ then

$$l_m = u_t \delta t$$

$$l_b = l_m \cot \theta + s_t \delta x$$

if $l_b + u_l \delta t > \delta x$

if $u_l \delta t \geq \delta x$

$$C_{cor2} = f_t$$

else

$$C_{cor2} = f_l - 0.5 (\delta y - u_t \delta t) (s_b \delta x + l_b + 2u_l \delta t - 2\delta x)$$

For C_{cor3} ,

if $f_b > 0$ and $f_l > 0$ then

$$l_m = \delta y - u_b \delta t$$

$$l_b = l_m \cot \theta + s_t \delta x$$

FIGURE 5: Schematic of fluid transport from the central cell i, j to the four oblique cells: (a) the position at time t^n , (b) the position at time t^n for case 1, (c) the position at time t^{n+1} for case 2, (d) the position at time t^{n+1} for case 3, and (e) the position at time t^{n+1} for case 4.

FIGURE 6: Continued.

FIGURE 6: Nine cases in which unit fluid may appear in calculating C_{cor1} for case III.

FIGURE 7: Parameters' definition for calculating C_{cor1} .

$$\text{if } u_l \delta t < \delta x$$

if $l_b > \delta x - u_l \delta t$

$$C_{\text{cor}3} = f_b - u_b \delta t (\delta x - u_l \delta t)$$

else

$$C_{\text{cor } 3} = f_l$$

else

$$C_{\text{cor } 3} = f_b$$

$$C_{\text{cor } 4} = u_b \delta t u_r \delta t$$

else

$$C_{\text{cor}4} = f_r - 1/2l_s(l_b + s_t\delta x)$$

else

$$C_{\text{cor } 4} = f_b$$

Similarly, the flux expressions of case I, case II, and case IV in four oblique directions can be solved.

For Ccor4,

if $f_b > 0$ and $f_r > 0$ then

$$l_m = \delta y - u_b \delta t$$

$$l_b = l_b \cot \theta + s_t \delta x$$

if $u_r \delta t < \delta x$

if $l_b \geq u_r \delta t$

(3) *Update the VOF Function.* After the fluid fluxes from the central cell (i, j) to eight adjacent cells during a time δt are calculated, the VOF function can be updated. Given fractions of the fluid volume in the central cell and the surrounding eight adjacent cells at time t are $F^t(i, j)$, $F^t-(i, j+1)$, $F^t(i, j-1)$, $F^t(i-1, j)$, $F^t(i+1, j)$, $F^t(i+1, j+1)$, $F^t(i-1,$

$j+1$), $F^t(i-1, j-1)$, and $F^t(i+1, j-1)$, respectively, at time $t+\delta t$, they are

$$F^{t+\delta t}(i, j) = \frac{F^t(i, j) - (C_t + C_b + C_l + C_r + C_{\text{cor}1} + C_{\text{cor}2} + C_{\text{cor}3} + C_{\text{cor}4})}{(\delta x \delta y)}, \quad (15)$$

$$\left. \begin{aligned} F^{t+\delta t}(i+1, j) &= F^t(i+1, j) + \frac{C_r}{(\delta x \delta y)} \\ F^{t+\delta t}(i, j+1) &= F^t(i, j+1) + \frac{C_t}{(\delta x \delta y)} \\ F^{t+\delta t}(i-1, j) &= F^t(i-1, j) + \frac{C_l}{(\delta x \delta y)} \\ F^{t+\delta t}(i, j-1) &= F^t(i, j-1) + \frac{C_b}{(\delta x \delta y)} \\ F^{t+\delta t}(i+1, j+1) &= F^t(i+1, j+1) + \frac{C_{\text{cor}1}}{(\delta x \delta y)} \\ F^{t+\delta t}(i-1, j+1) &= F^t(i-1, j+1) + \frac{C_{\text{cor}2}}{(\delta x \delta y)} \\ F^{t+\delta t}(i-1, j-1) &= F^t(i-1, j-1) + \frac{C_{\text{cor}3}}{(\delta x \delta y)} \\ F^{t+\delta t}(i+1, j-1) &= F^t(i+1, j-1) + \frac{C_{\text{cor}4}}{(\delta x \delta y)} \end{aligned} \right\}. \quad (16)$$

Repeat the calculation for each cell; then, the update for the VOF function in the whole field can be completed.

4. Case Studies

The expansive diffusion process of expansible grout in a fracture was tracked to validate the precision of the unsplit Youngs algorithm. The results were compared with the analytical solution and the traditional split Youngs method.

4.1. Calculation Model. The large-size and well-developed fracture can be modeled as a parallel-plate fracture. Four side edges of the fracture are set as the pressure outlet, as shown in Figure 8. The length and width of the fracture are both 1.6 m and the fracture aperture is 3 mm. A grouting hole is located in the center of the model, through which a certain amount of expansible grout is injected into the fracture. Assuming that the fracture is initially filled with air, the injected grout will partially displace it expansible and occupy a circle area with the radius R_0 . According to Mitani and Hamada [23], the grout density changes with time as below:

$$\rho_t = \frac{\rho_0}{1 + \alpha t}, \quad (17)$$

where ρ_0 is the initial grout density with a value of 1100 kg/m³, α is the expansion rate, t is the time, and ρ_t is the grout density at time t . The grout viscosity remains unchanged at 1.0 pa·s before gelation.

Considering the fracture model is geometrical symmetry, the solving could be simplified by only calculating a quarter of the simulated region, as shown in Figure 9. Therefore, there are two symmetry boundaries and the other boundary conditions are pressure outlet.

The grouting quantity is represented by the initial diffusion radius of expansible grout in a fracture. The grouts with different components have different expansion rates. The input parameters are shown in Table 1. Two types of expansible grout with different initial radius were used to validate the unsplit Youngs algorithm.

The analytical solution of the diffusion distance of the expansible grout in a fracture is expressed as follows [24]:

$$R_t = R_0 \sqrt{1 + \alpha t}, \quad (18)$$

where R_0 is the initial diffusion radius and R_t is the diffusion radius at time t .

4.2. Numerical Calculation. The diffusion behavior of the expansible grout can be described by one continuity equation and three momentum conservation equations. The continuity equation is given as

FIGURE 8: Fracture model.

FIGURE 9: Calculation region.

TABLE 1: Input material parameters.

Materials	Density (kg/m ³)	Viscosity (Pa·s)	Expansion rate (s ⁻¹)	Initial diffusion radius (m)
Air	1.205	1.8×10^{-5}	—	—
Grout	1100 (initial)	1.0	0.1	0.25
			0.1	0.55
			0.2	0.25
			0.2	0.55

$$\frac{\partial(\rho_t F)}{\partial t} + \nabla \cdot (\rho_t F \mathbf{U}) = 0, \quad (19)$$

where F is the volume fraction of the expansible grout and $\mathbf{U}$ is the velocity vector.

The two phases of grout and air share one single set of momentum equations, which are written as follows:

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot [\mu(\nabla \mathbf{U} + \nabla \mathbf{U}^T)] + \rho g, \quad (20)$$

where g is the gravity acceleration, ρ and μ are the volume fraction weighted average of the density and viscosity, respectively, and p is the pressure.

The solution domain was meshed into 6400 hexahedral elements with an edge length of 0.01 m. Based on the finite volume method, the governing equations were solved iteratively by the Semi-Implicit Method for Pressure Linked Equations algorithm (SIMPLE). The procedure is listed below:

- (1) Define the boundary conditions and initialize the fractional volume of each cell, velocity vector, and pressure
- (2) Determine the material properties of expansible grout according to the time and the VOF function
- (3) Solve the discretized momentum equation to obtain the current velocity
- (4) Solve the pressure correction equation to obtain the pressure correction value
- (5) Update the current velocity and pressure
- (6) Repeat Steps (3)–(5) until the flow field converges
- (7) Use the unsplit Youngs algorithm to update the VOF function and advance the interface

Return to step (2), update the time, and the numerical process proceeds to the next iteration. The numerical calculation was programmed by FORTRAN. The procedure is shown in Figure 10.

FIGURE 10: Numerical procedure.

4.3. Comparison and Discussion. Since different grouting quantities and expansion rates lead to different diffusion rates, in Cases 1–4, the grout takes different times to reach the boundary. Considering the limitation of the simulation domain, the calculation time is set as 240 s, 100 s, 55 s, and 24 s for Cases 1–4.

Figures 11–14 are cloud charts of the diffusion process of the expansible grout in Case 1 ($t = 80$ s, 160 s, 240 s), Case 2 ($t = 20$ s, 60 s, 100 s), Case 3 ($t = 15$ s, 35 s, 55 s), and Case 4 ($t = 8$ s, 16 s, 24 s), respectively. It shows that the grout diffusion rate is quite different under different conditions of the initial filling range and the grout expansion rates. However, centered on the grouting hole, the grout is circular diffusion and spreads symmetrically towards the surroundings, which is consistent with the theoretical diffusion pattern. The grout diffusion distance calculated by the unsplit Youngs algorithm is in good agreement with the analytical solution. The contour of the grout front calculated by the two methods almost completely coincides. However, the calculation result from the split algorithm brings a significant deviation from the analytic solution. The diffusion ranges calculated by the split algorithm at different times lag behind the theoretical results. The grout contour calculated by the split algorithm always lies inside the analytical solution, and the deviation increases gradually with time.

The average diffusion radius at different times was calculated for Cases 1–4, shown in Tables 2–5, and plotted in Figures 15(a)–15(d), respectively. Obviously, the diffusion radius calculated by the unsplit Youngs algorithm always agrees well with the analytical solution with the increasing time, and their contours of the grout front almost coincide. From Case 1 to Case 4, the maximum absolute errors are 0.92 cm, 1.1 cm, 0.41 cm, and 0.26 cm, respectively. By contrast, the diffusion radius calculated by the split algorithm is always smaller than the analytical solution, and the deviation gradually increases with time. The maximum absolute errors are 10.71 cm, 7.99 cm, 4.05 cm, and 4.13 cm, respectively.

Figure 16 shows the relative error between the numerical method and analytical solution with respect to the diffusion radius under the four cases. The maximum relative errors calculated by the unsplit algorithm are 2.14% and 2.37% and 1.21% and 0.85%, respectively, while they are 13.71%, 10.68%, 5.27%, and 5.48% calculated by the split algorithm, respectively, which are 6.4 times, 4.5 times, 4.36 times, and 6.45 times, respectively, higher than the unsplit algorithm. Besides, it can be observed that the relative error of the split algorithm is gradually increasing with the increase of the

FIGURE 11: Numerical results of grout spread, shape, and range of Case 1. (a) Numerical results of the split algorithm at $t = 80$ s, 160 s, and 240 s. (b) Numerical results of unsplit algorithm at $t = 80$ s, 160 s, and 240 s.

FIGURE 12: Continued.

FIGURE 12: Numerical results of grout spread, shape, and range of Case 2. (a) Numerical results of split algorithm at $t = 20$ s, 60 s, and 100 s (b) Numerical results of unsplit algorithm at $t = 20$ s, 60 s, and 100 s.

FIGURE 13: Numerical results of grout spread, shape, and range of Case 3. (a) Numerical results of the split algorithm at $t = 15$ s, 35 s, and 55 s. (b) Numerical results of unsplit algorithm at $t = 15$ s, 35 s, and 55 s.

diffusion radius, while the relative error of the unsplit algorithm declines slightly.

The above comparison of relative errors shows that the numerical results of the split Youngs algorithm deviate from the analytical solution more and more as the diffusion

distance increases. Because the split Youngs algorithm does not consider fluxes in the diagonal direction, the arithmetic errors will be accumulated. By comparison, the unsplit Youngs algorithm better guarantees the conservation of fluid mass in the flow by adding the fluxes in the diagonal

FIGURE 14: Numerical results of grout spread, shape, and range of Case 4. (a) Numerical results of the split algorithm at $t = 8$ s, 16 s, and 24 s. (b) Numerical results of unsplit algorithm at $t = 8$ s, 16 s, and 24 s.

TABLE 2: Calculation results of grout spread radius at different times in case 1.

Time (s)	Analytical (m)	Split algorithm			Unsplit algorithm		
		Numerical results (m)	Absolute error (m)	Relative error (%)	Numerical results (m)	Absolute error (m)	Relative error (%)
20	0.2449	0.2233	0.0217	8.85	0.2397	0.0052	2.14
40	0.3317	0.2974	0.0343	10.34	0.3251	0.0065	1.97
60	0.4000	0.3547	0.0453	11.33	0.3923	0.0077	1.91
80	0.4583	0.4039	0.0544	11.86	0.4503	0.0080	1.75
100	0.5099	0.4479	0.0620	12.16	0.5016	0.0083	1.63
120	0.5568	0.4874	0.0694	12.46	0.5477	0.0091	1.63
140	0.6000	0.5240	0.0760	12.67	0.5909	0.0091	1.51
160	0.6403	0.5583	0.0820	12.81	0.6312	0.0091	1.42
180	0.6782	0.5906	0.0876	12.92	0.6688	0.0095	1.40
200	0.7141	0.6208	0.0934	13.08	0.7045	0.0096	1.35
220	0.7483	0.6499	0.0984	13.15	0.7391	0.0093	1.24
240	0.7810	0.6740	0.1071	13.71	0.7718	0.0092	1.18

direction. This reduces the calculation error effectively and improves the accuracy of tracking the moving interface significantly.

It should be noted that the present algorithm is still complex. There are four types of interface when it locates

in the cell. For each type, fluid transport could appear in eight directions. Each fluid flux from the central cell to four oblique direction cells could have nine cases. In addition, the proposed method can only be used to solve two-dimensional interface tracking. The three-

TABLE 3: Calculation results of grout spread radius at different times in case 2.

Time (s)	Analytical results (m)	Split algorithm			Unsplit algorithm		
		Numerical results (m)	Absolute error (m)	Relative error (%)	Numerical results (m)	Absolute error (m)	Relative error (%)
10	0.2550	0.2373	0.0176	6.90	0.2489	0.0060	2.37
20	0.3464	0.3174	0.0290	8.37	0.3391	0.0073	2.10
30	0.4183	0.3803	0.0381	9.10	0.4099	0.0084	2.01
40	0.4796	0.4334	0.0462	9.64	0.4704	0.0091	1.91
50	0.5339	0.4808	0.0531	9.95	0.5243	0.0096	1.80
60	0.5831	0.5239	0.0592	10.15	0.5731	0.0100	1.72
70	0.6285	0.5634	0.0651	10.35	0.6182	0.0103	1.64
80	0.6708	0.6005	0.0703	10.49	0.6603	0.0105	1.57
90	0.7106	0.6351	0.0755	10.63	0.6999	0.0107	1.51
100	0.7483	0.6684	0.0799	10.68	0.7373	0.0110	1.47

TABLE 4: Calculation results of grout spread radius at different times in case 3.

Time (s)	Analytical results (m)	Split algorithm			Unsplit algorithm		
		Numerical results (m)	Absolute error (m)	Relative error (%)	Numerical results (m)	Absolute error (m)	Relative error (%)
5	0.3000	0.2927	0.0073	2.44	0.2964	0.0036	1.21
10	0.3742	0.3612	0.0130	3.47	0.3707	0.0035	0.93
15	0.4359	0.4187	0.0172	3.94	0.4318	0.0041	0.94
20	0.4899	0.4691	0.0208	4.24	0.4860	0.0039	0.79
25	0.5385	0.5140	0.0245	4.55	0.5345	0.0040	0.74
30	0.5831	0.5553	0.0278	4.76	0.5798	0.0033	0.57
35	0.6245	0.5940	0.0305	4.89	0.6210	0.0035	0.56
40	0.6633	0.6302	0.0332	5.00	0.6602	0.0032	0.48
45	0.7000	0.6640	0.0360	5.15	0.6966	0.0034	0.48
50	0.7348	0.6964	0.0385	5.23	0.7314	0.0035	0.48
55	0.7681	0.7276	0.0405	5.27	0.7646	0.0035	0.46

TABLE 5: Calculation results of grout spread radius at different times in Case 4.

Time (s)	Analytical results (m)	Split algorithm			Unsplit algorithm		
		Numerical results (m)	Absolute error (m)	Relative error (%)	Numerical results (m)	Numerical results (m)	Absolute error (%)
2	0.2898	0.2831	0.0067	2.33	0.2874	0.0025	0.85
4	0.3578	0.3460	0.0117	3.28	0.3556	0.0021	0.60
6	0.4147	0.3989	0.0159	3.83	0.4121	0.0026	0.63
8	0.4648	0.4450	0.0197	4.24	0.4634	0.0013	0.28
10	0.5099	0.4866	0.0233	4.57	0.5088	0.0011	0.21
12	0.5514	0.5252	0.0261	4.74	0.5503	0.0011	0.20
14	0.5899	0.5608	0.0291	4.94	0.5873	0.0026	0.44
16	0.6261	0.5944	0.0317	5.07	0.6236	0.0025	0.40
18	0.6603	0.6258	0.0345	5.22	0.6584	0.0019	0.29
20	0.6928	0.6558	0.0370	5.34	0.6910	0.0018	0.26
22	0.7239	0.6849	0.0389	5.38	0.7225	0.0014	0.20
24	0.7537	0.7123	0.0413	5.48	0.7524	0.0012	0.17

dimensional Youngs method has to take into account the fluid flux in the z -direction, which needs to reprogram algorithm. This paper is our preliminary result based on

the idea of the unsplit interface. We will further develop the unsplit algorithm of the moving interface to make it more widely applicable.

FIGURE 15: Comparison of grout spread radius at different times: (a) $R_0 = 0.1$ m, $\alpha = 0.25$; (b) $R_0 = 0.1$ m, $\alpha = 0.55$; (c) $R_0 = 0.2$ m, $\alpha = 0.25$; (d) $R_0 = 0.2$ m, $\alpha = 0.55$.

FIGURE 16: Continued.

FIGURE 16: Relative error with respect to different grout spread radii: (a) $R_0 = 0.1$ m, $\alpha = 0.25$; (b) $R_0 = 0.1$ m, $\alpha = 0.55$; (c) $R_0 = 0.2$ m, $\alpha = 0.25$; (d) $R_0 = 0.2$ m, $\alpha = 0.55$.

5. Conclusion

How to track the moving interface accurately is crucial to predict the grout diffusion behavior. Considering the fluxes from a central cell to oblique adjacent ones, unsplit interface advancement formulas were deduced under the structured cell system. Based on the traditional split Youngs algorithm, the unsplit algorithm improves the accuracy of tracking the moving interface by introducing the basic principle of the direction unsplit method. The diffusion process of expandable grout in a parallel-plate fracture is studied by the proposed algorithm and split Youngs algorithm. The results show that the diffusion distance calculated by the unsplit Youngs algorithm agrees better with the analytical solution under different cases. Compared with the traditional split Youngs algorithm, the new method can reduce the calculation error effectively and improve the tracking accuracy of the grout front significantly.

This study can be used to predict the grout front in a fracture, so the arrangement of grouting hole can be designed reasonably. It has been used in the repair and maintenance of transportation infrastructure, e.g., repair for cavity beneath cement concrete pavement. On this basis, we will try to apply the unsplit Youngs algorithm to the tracking of the grout moving interface in fracture network in the future; thus, it is likely to capture the grout diffusion process in more complex geology.

Data Availability

All data in our study are available from the corresponding author by request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors sincerely appreciate the editor for critical comments. This research was supported by the National Key Research and Development Plan (Grant nos.

2017YFC0504503 and 2018YFB1600200), Program for Science and Technology Innovation Talents in Universities of Henan Province (Grant no. 19HASTIT041), National Natural Science Foundation of China (Grant nos. 51908514, 51878624, 51878622 and U1404525), Natural Science Foundation of Henan Province (Grant no. 202300410746), Postdoctoral Research Sponsorship in Henan Province (Grant no. 19030026), Key Research Projects of Higher Education in Henan Province (Grant no. 18A580001), Program for Innovative Research Team (in Science and Technology) at the University of Henan Province (Grant no. 18IRTSTHN007), and Major Scientific and Technological Special Project in Henan (Grant no. 181100310400).

References

- [1] J. Funehag and Å. Fransson, "Sealing narrow fractures with a newtonian fluid: model prediction for grouting verified by field study," *Tunnelling and Underground Space Technology*, vol. 21, no. 5, pp. 492–498, 2006.
- [2] T.-L. Chen, L.-Y. Zhang, and D.-L. Zhang, "An FEM/VOF hybrid formulation for fracture grouting modelling," *Computers and Geotechnics*, vol. 58, pp. 14–27, 2014.
- [3] S. Briggs, B. W. Karney, and B. E. Sleep, "Numerical modelling of flow and transport in rough fractures," *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 6, no. 6, pp. 535–545, 2014.
- [4] A. Naudts, "Irreversible changes in the grouting industry caused by polyurethane grouting: an overview of 30 years of polyurethane grouting," in *Grouting and Ground Treatment*, pp. 1266–1280, American Society of Civil Engineers, Reston, VI, USA, 2003.
- [5] F. Wang, "Directional fracture grouting method with polymer for seepage control of dikes and dams," 2012.
- [6] C. C. Guo, "A field experimental study on the diffusion behavior of expanding polymer grouting material in soil," *Soil Mechanics and Foundation Engineering*, vol. 56, no. 3, pp. 171–177, 2019.
- [7] Z. Q. Feng and H. P. Kang, "Development and application of new waterproof grouting materials of polyurethane," *Chinese Journal of Geotechnical Engineering*, vol. 32, no. 3, pp. 375–380, 2010.

- [8] M. S. Shi, *Research on Polymer Grouting Material Properties and Directional Fracturing Grouting Mechanism for Dykes and Dams*, Dalian University of Technology, Dalian, China, 2010.
- [9] L. Hässler, U. Håkansson, and H. Stille, "Computer-simulated flow of grouts in jointed rock," *Tunnelling and Underground Space Technology*, vol. 7, no. 4, pp. 441–446, 1992.
- [10] M. Eriksson, H. Stille, and J. Andersson, "Numerical calculations for prediction of grout spread with account for filtration and varying aperture," *Tunnelling and Underground Space Technology*, vol. 15, no. 4, pp. 353–364, 2000.
- [11] M. J. Yang, Z. Q. Yue, P. K. Lee, B. Su, and L. G. Tham, "Prediction of grout penetration in fractured rocks by numerical simulation," *Canadian Geotechnical Journal*, vol. 39, no. 6, pp. 1384–1394, 2002.
- [12] J. Liu, "Diffusion law model test and numerical simulation of cement fracture grouting," *Chinese Journal of Rock Mechanics and Engineering*, vol. 31, no. 12, pp. 2445–2452, 2012.
- [13] P. Zhao, *The Diffusion Law of Cross Fracture Grouting Based on the Time-dependent Behavior of Viscosity of Grout*, Shandong University, Jinan, China, 2016.
- [14] X. L. Li, "Application of youngs method in tracking the moving interface of expansible grout," *Rock and Soil Mechanics*, vol. 38, no. 12, pp. 3491–3497, 2017.
- [15] X. Li, "A quasi-3D numerical model for grout injection in a parallel fracture based on finite volume method," *Complexity*, vol. 2019, Article ID 4139616, 18 pages, 2019.
- [16] Y. Y. Hu, K. Q. Zhu, and B. S. Xi, "Unsplit youngs' algorithm for the fractional volume function," *Journal of Tsinghua University*, vol. 42, no. 2, pp. 232–234, 2002.
- [17] C. Liu, T. B. Ma, and J. G. Ning, "Unsplit transport method based on youngs' interface reconstruction algorithm in eulerian method," in *Proceedings of the 5th International Conference on Nonlinear Mechanics*, pp. 199–203, Marrakech, Morocco, June 2007.
- [18] R. R. Rao, L. A. Mondy, D. R. Noble, H. K. Moffat, D. B. Adolf, and P. K. Notz, "A level set method to study foam processing: a validation study," *International Journal for Numerical Methods in Fluids*, vol. 68, no. 11, pp. 1362–1392, 2012.
- [19] Y. Liu, N. Ma, and X. Gu, "3-D simulation of large amplitude liquid sloshing using youngs-VOF method," *Journal of Harbin Engineering University*, vol. 33, no. 9, pp. 1075–1078, 2012.
- [20] M. Rudman, "Volume-tracking methods for interfacial flow calculations," *International Journal for Numerical Methods in Fluids*, vol. 24, no. 7, pp. 671–691, 1997.
- [21] C. W. Hirt and B. D. Nichols, "Volume of fluid (vof) method for the dynamics of free boundaries," *Journal of Computational Physics*, vol. 39, no. 1, pp. 201–225, 1981.
- [22] D. L. Youngs, *Time-Dependent Multi-Material Flow with Large Fluid Distortion*, New York University, New York City, NY, USA, 1982.
- [23] T. Mitani and H. Hamada, "Prediction of flow patterns in the polyurethane foaming process by numerical simulation considering foam expansion," *Polymer Engineering & Science*, vol. 43, no. 9, pp. 1603–1612, 2003.
- [24] X. L. Li, "Diffusion model of an ideal expansible grout in single fracture," *Chinese Journal of Rock Mechanics and Engineering*, vol. 37, no. 5, pp. 1207–1217, 2018.