

Article

Three-Dimensional Streamline Tracing Method over Tetrahedral Domains

Yi Luo ¹, Liehui Zhang ¹, Yin Feng ^{2,*} and Yulong Zhao ^{1,*}

¹ State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation, Southwest Petroleum University, Chengdu 610500, China; luoyi495215232@126.com (Y.L.); zhangliehui@swpu.edu.cn (L.Z.)

² Department of Petroleum Engineering, University of Louisiana at Lafayette, Lafayette, LA 70504, USA

* Correspondence: yin.feng@louisiana.edu (Y.F.); swpuzhao@swpu.edu.cn (Y.Z.)

Received: 17 October 2020; Accepted: 16 November 2020; Published: 18 November 2020



Abstract: Getting a clear understanding of the fluid velocity field in underground porous media is critical to various engineering applications, such as oil/gas reservoirs, CO₂ sequestration, groundwater, etc. As an effective visualization tool and efficient transport behaviors solution algorithm, the streamline-based method was improved significantly by numerous studies conducted in the last couple of decades. However, the implementation of streamline simulation is still challenging while working with Finite Element Method (FEM) over 3D tetrahedral domains, where the mass conservation is not guaranteed. Considering the increased computational cost to enforce mass conservation in FEM and additional complexity, a new three-dimensional streamline tracing algorithm is presented that only relies on the velocity vector of a flow field on each vertex of a tetrahedron in a 3D unstructured mesh system. Owing to the shape functions and transformation equations between the master element and actual element, the exit coordinate leaving a tetrahedral element can be determined effectively. As a result, Time of Flight (TOF), the coordinate variable along each streamline, can be calculated accurately and efficiently because that the analytical solution depicting the trajectory in Master Element is deduced. The presented streamline-based method is tested under FEniCS, a programming framework for FEM, which eases the implementation and further development of the presented method.

Keywords: 3D streamline tracing; tetrahedral element; streamline-based simulation; Finite Element Method; FEniCS

1. Introduction

The dynamic physical processes of fluid flow and transport in underground porous media play an important role in a lot of applications, i.e., fast history matching [1,2], enhanced oil recovery [3–7], polymer flooding simulation [8], fractured reservoir simulation [9,10], CO₂ storage [11,12], and solving the solute transport and reactive transport in underground porous media [13–19]. The streamline-based simulation can describe dynamic physical processes of fluid flowing and transporting in underground porous media.

Due to the application of numerical theories such as stream-tube concept, image-of-well, and the superposition principle [20–22], the streamline method has been developing since the 1950s in the petroleum industry, which can trace the fluid pathway. After the introduction of the new concept of “Time Of Flight” (TOF, denoted symbolically by τ) introduced in streamline simulation, the modern streamline simulation concept has been presented and studied by many researchers [7,23,24] since the 1990s. From then on, the streamline simulation is not just used to visualize a flow field tool but to solve transport problems in complex domains. One feature of streamline-based simulation is that a complex 3D fluid-flow simulation could be decoupled into a series of 1D solutions along

each streamline, which makes the streamline-based simulation efficient when dealing with large, heterogeneous reservoir and complex enhanced oil recovery (EOR) processes. Some principles for most streamline simulations [25,26] are shown as follows: (1) tracing streamlines and constructing a 1D coordinate system according to the concept of TOF; (2) updating streamlines; (3) using TOF to express the mass equations; and (4) solving the 1D formulation of the transport equation along each streamline in terms of TOF.

The streamline tracing algorithm plays an important role in streamline-based simulation. One category of such algorithm is to use numerically methods such as Runge–Kutta-type solvers to trace streamlines. These methods are applicable if all we need is to generally visualize flow pathways. However, to solve the transport behaviors along these streamlines (known as the concept of modern streamline stimulation), Runge–Kutta-type methods are not acceptable, because these methods are not accurate or efficient enough for 3D large-scale domains and very difficult to construct the TOF grid system [27]. The other category is semi-analytical approaches, which are a good choice for tracing streamlines for the presented purpose. In early streamline-based simulation applications, Lin [22] adopted the image-well method to trace streamlines in a 2D reservoir, which is not robust to determine the positions and numbers through a trial-and-error procedure.

Then, Numere [28] built a streamline system via the Boundary Element Method, which can address the problem of the image-well method, but it is difficult to be implemented in a 3D and heterogeneous computation domain. Fortunately, Pollock [29] has presented a very famous semi-analytical streamline tracing approach, assuming that the velocity in all directions is linearly distributed and does not affect each other. In addition, the concept of time of flight (TOF) has been introduced, which is efficient and widely applied to a lot of streamline-based simulation in 3D reservoirs [24,27,30–35]. The Pollock method is widely used to establish streamlines over structured grids. Prevost et al. [30] extended the Pollock method to finite-volume flow solutions using distorted quadrilaterals (2-D) or hexahedra (3-D). Although the Pollock method has many advantages in tracing streamlines, the drawbacks are obvious. For instance, it relies on conservative flux at each facet of a cell, and it could not trace streamlines in a mesh system when the cells contain sink or source [26,29,36].

Although most of these streamline tracing methods have been applied to structured meshes, some researches on tracing streamlines in unstructured meshes has been presented [27,31–34,36–47]. There exist some drawbacks in those algorithms, which are listed below:

Firstly, it is required that the flux must be conservative flux and the flux must belong to each facet of an element during the tracing streamline process.

Secondly, all the formulations of tracing the streamlines should depend on a control volume.

Thirdly, during the tracing streamline process, another numerical solver is needed to solve the Ordinary Differential Equations (ODEs). Therefore, the TOFs could not be computed efficiently, which limits the implementation of the streamline-based simulation.

Fourthly, this type of streamline tracing method is just a visualization tool for the fluid flow field and could not implement streamline-based simulation.

Fifth, although there are some presented methods for streamline tracing methods in 2D and 3D unstructured mesh system such as triangle mesh [31,39,44,47], Perpendicular Bisector (PEBI) cell [32], and fully unstructured 3D Voronoi grids [45,46], to the best knowledge of the authors, few research studies on tracing streamlines over tetrahedral domains (created by FEM) are based on the velocity vector on each vertex.

The objectives of this paper are, on one hand, to present a new three-dimensional streamline tracing method based on the flow field in unstructured tetrahedron mesh systems where the conservative flux is not necessary; on the other hand, the presented work is implemented and tested in FEniCS; when the flow solver runs in FEniCS, only the finite element mesh and the resultant velocity vector are needed. The advances and features of the new method are summarized as follows.

- (1) During the process of streamline tracing, only the coordinate and velocity vector on the vertex are acquired. Neither flux on element facets nor the flux conservation within an element need to be considered.
- (2) For any given coordinate of an entrance point in an unstructured tetrahedron element, the corresponding exit coordinate can be determined.
- (3) The time of flight can be calculated accurately and efficiently because the analytical solution that could describe the pathway of a particle in an unstructured tetrahedron element is deduced.

2. Mathematics Fundamentals for Tracing Streamlines over Tetrahedron Grid System

To trace streamlines over the tetrahedron grid system created by the Finite Element Method (FEM), the mathematics fundamentals are summarized as following sections.

2.1. Fundamentals of Shape Function

In this paper, the main fundament for tracing streaming over the unstructured tetrahedron mesh is the shape function in the Master Element (the Master Element schematic is shown as Figure 1b), which is used to evaluate value through the known values, which acquired on the tetrahedron vertices. The values need to interpolate in this study are coordinates and velocity vectors. The shape function used in this study is the 1st-order Lagrange function, which yields Equation (1). For any variable (U), belonging at point (ξ, η, ζ) could be evaluated by Equation (2), when the values U_1, U_2, U_3 , and U_4 at tetrahedron vertex $P_1(0,0,0), P_2(1,0,0), P_3(0,1,0)$ and $P_4(1,1,1)$ are acquired.

$$\begin{aligned} S_1(\xi, \eta, \zeta) &= a_1 + b_1\xi + c_1\eta + d_1\zeta \\ S_2(\xi, \eta, \zeta) &= a_2 + b_2\xi + c_2\eta + d_2\zeta \\ S_3(\xi, \eta, \zeta) &= a_3 + b_3\xi + c_3\eta + d_3\zeta \\ S_4(\xi, \eta, \zeta) &= a_4 + b_4\xi + c_4\eta + d_4\zeta \end{aligned} \quad (1)$$

$$U(\xi, \eta, \zeta) = S_1(\xi, \eta, \zeta)U_1 + S_2(\xi, \eta, \zeta)U_2 + S_3(\xi, \eta, \zeta)U_3 + S_4(\xi, \eta, \zeta)U_4 \quad (2)$$

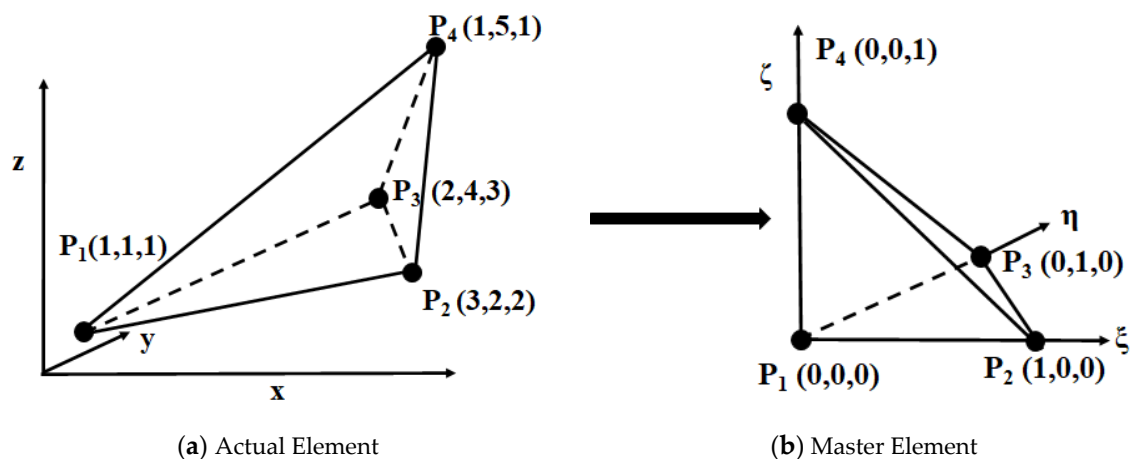


Figure 1. The transform process schematic for coordinate of the Actual Element and Master Element where (a) represents tetrahedron in the Actual Element and (b) represents the tetrahedron in the Master Element.

Taking vertex coordinates $P_1(0,0,0)$, $P_2(1,0,0)$, $P_3(0,1,0)$, and $P_4(0,0,1)$ as U_1 , U_2 , U_3 and U_4 , respectively, the result is shown as Equation (3), and the shape function is given by Equation (4), which is solved from Equations (1) and (2).

$$\begin{aligned} S_1(0,0,0) &= 1; S_2(0,0,0) = 0; S_3(0,0,0) = 0; S_4(0,0,0) = 0 \\ S_1(1,0,0) &= 0; S_2(1,0,0) = 1; S_3(1,0,0) = 0; S_4(1,0,0) = 0 \\ S_1(0,1,0) &= 0; S_2(0,1,0) = 0; S_3(0,1,0) = 1; S_4(0,1,0) = 0 \\ S_1(0,0,1) &= 0; S_2(0,0,1) = 0; S_3(0,0,1) = 0; S_4(0,0,1) = 1 \end{aligned} \quad (3)$$

$$\begin{aligned} S_1(\xi, \eta, \zeta) &= 1 - \xi - \eta - \zeta \\ S_2(\xi, \eta, \zeta) &= \xi \\ S_3(\xi, \eta, \zeta) &= \eta \\ S_4(\xi, \eta, \zeta) &= \zeta \end{aligned} \quad (4)$$

2.2. Transformation Formulation of Coordinates and Velocity Vectors between Actual and Master Element

Arbitrary shape tetrahedron elements in Actual Element (x - y - z plane) are shown in Figure 1a, the geometrical is very complex for tracing a streamline, so it is very important to minimize the geometrical complexity. Therefore, tetrahedron elements in Actual Element need to transform into Master Element (ξ - η - ζ plane) where the geometrical complexity is minimized, as shown in Figure 1b, and the transform process schematic from Actual Element to Master Element is shown as Figure 1.

The first variable needed to transform between Master and Actual Elements is the coordinates. The coordinates in Actual Elements could transform to Master Elements easily by Equation (2), which only needs to take coordinates (x, y, z) into Equation (2) and replace U_i ; the result is shown in Equation (5). The (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) and (x_4, y_4, z_4) , are the coordinates of four vertexes of the tetrahedron in the Actual Element.

$$\begin{aligned} x &= (1-\xi-\eta-\zeta)x_1 + \xi x_2 + \eta x_3 + \zeta x_4 \\ y &= (1-\xi-\eta-\zeta)y_1 + \xi y_2 + \eta y_3 + \zeta y_4 \\ z &= (1-\xi-\eta-\zeta)z_1 + \xi z_2 + \eta z_3 + \zeta z_4 \end{aligned} \quad (5)$$

Then, transformation coordinates (x, y, z) in Actual Elements to coordinates (ξ, η, ζ) in Master Element is more complex. The transformation formulation is shown as Equation (6), which is derived from Equation (5).

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} x_1 - x_2x_1 - x_3x_1 - x_4 \\ y_1 - y_2y_1 - y_3y_1 - y_4 \\ z_1 - z_2z_1 - z_3z_1 - z_4 \end{bmatrix}^{-1} \begin{bmatrix} x_1 - x \\ y_1 - y \\ z_1 - z \end{bmatrix} \quad (6)$$

Another variable needed to transform between Actual and Master Elements is the velocity vector. For any given velocity vector $[v_x, v_y, v_z]$ at a point (x, y, z) in an Actual Element, it needs to be transformed to the Master Element ($[v_\xi, v_\eta, v_\zeta]$). According to Equation (6), the transformation formula for velocity vectors from an Actual Element to a Master Element is expressed as shown in Equation (7).

$$\begin{bmatrix} v_\xi \\ v_\eta \\ v_\zeta \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = - \begin{bmatrix} x_1 - x_2x_1 - x_3x_1 - x_4 \\ y_1 - y_2y_1 - y_3y_1 - y_4 \\ z_1 - z_2z_1 - z_3z_1 - z_4 \end{bmatrix}^{-1} \begin{bmatrix} v_{x1} \\ v_{y1} \\ v_{z1} \end{bmatrix} \quad (7)$$

2.3. The Analytical Solution for Describing the Trajectory in Master Element

In order to obtain the analytical solution for describing the trajectory in Master Element, it needs to know the motion equations at any point (ξ, η, ζ) , that is to say, the velocity vector for any point is the key point. According to the shape function in the Master Element, the velocity vector of any

point yields Equation (8), when the velocity vectors ($[v_{\xi 1}, v_{\eta 1}, v_{\zeta 1}]$, $[v_{\xi 2}, v_{\eta 2}, v_{\zeta 2}]$, $[v_{\xi 3}, v_{\eta 3}, v_{\zeta 3}]$, and $[v_{\xi 4}, v_{\eta 4}, v_{\zeta 4}]$) at each vertex are obtained.

$$\begin{aligned} v_{\xi} &= \frac{d\xi}{dt} = v_{\xi 1} - (v_{\xi 1} - v_{\xi 2})\xi - (v_{\xi 1} - v_{\xi 3})\eta - (v_{\xi 1} - v_{\xi 4})\zeta \\ v_{\eta} &= \frac{d\eta}{dt} = v_{\eta 1} - (v_{\eta 1} - v_{\eta 2})\xi - (v_{\eta 1} - v_{\eta 3})\eta - (v_{\eta 1} - v_{\eta 4})\zeta \\ v_{\zeta} &= \frac{d\zeta}{dt} = v_{\zeta 1} - (v_{\zeta 1} - v_{\zeta 2})\xi - (v_{\zeta 1} - v_{\zeta 3})\eta - (v_{\zeta 1} - v_{\zeta 4})\zeta \end{aligned} \quad (8)$$

The initial conditions are shown in Equation (9).

$$\begin{aligned} \xi(t=0) &= \xi_0; \eta(t=0) = \eta_0; \zeta(t=0) = \zeta_0 \\ \frac{d\xi}{dt}(t=0) &= v_{\xi 0}; \frac{d\eta}{dt}(t=0) = v_{\eta 0}; \frac{d\zeta}{dt}(t=0) = v_{\zeta 0} \end{aligned} \quad (9)$$

Then, the matrix format of Equation (8) is described as shown in Equation (10):

$$\frac{dX}{dt} = B + AX$$

$$X = \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}, B = \begin{pmatrix} v_{\xi 1} \\ v_{\eta 1} \\ v_{\zeta 1} \end{pmatrix}, A = \begin{pmatrix} -(v_{\xi 1} - v_{\xi 2}) & -(v_{\xi 1} - v_{\xi 3}) & -(v_{\xi 1} - v_{\xi 4}) \\ -(v_{\eta 1} - v_{\eta 2}) & -(v_{\eta 1} - v_{\eta 3}) & -(v_{\eta 1} - v_{\eta 4}) \\ -(v_{\zeta 1} - v_{\zeta 2}) & -(v_{\zeta 1} - v_{\zeta 3}) & -(v_{\zeta 1} - v_{\zeta 4}) \end{pmatrix} \quad (10)$$

According to Gregory et al. [48], the general solution of Equation (11) is of the form

$$X(t) = E_1\Phi_1(t) + E_2\Phi_2(t) + E_3\Phi_3(t) + C \quad (11)$$

where the particular function $\Phi_1(t)$, $\Phi_2(t)$ and $\Phi_3(t)$, and coefficients E_1 , E_2 , E_3 , and C depend on the eigenvalues of A . There are nine separate cases that $X(t)$ takes depending on different combinations eigenvalues. Here is Case 1 (Equation (12)), which means A has three nonzero real eigenvalues ($0 \neq r_1 \neq r_2 \neq r_3 \neq 0$), and the other cases are shown in Appendix A.

(1) Case 1: A has three nonzero real eigenvalues ($0 \neq r_1 \neq r_2 \neq r_3 \neq 0$)

$$\begin{aligned} X(t) &= E_1e^{r_1 t} + E_2e^{r_2 t} + E_3e^{r_3 t} + X_c \\ E_1 &= \left(\frac{A-r_2 I}{r_1-r_2} \right) \left(\frac{r_3 I - A}{r_3-r_1} \right) (X_0 - X_c) \\ E_2 &= \left(\frac{A-r_3 I}{r_2-r_3} \right) \left(\frac{r_1 I - A}{r_1-r_2} \right) (X_0 - X_c) \\ E_3 &= \left(\frac{A-r_1 I}{r_3-r_1} \right) \left(\frac{r_2 I - A}{r_2-r_3} \right) (X_0 - X_c) \\ X_c &= -BA^{-1}, X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (12)$$

2.4. Computing the TOF and Exit Coordinate

For any particle in the Mater Element whose start point is P_0 , it will leave tetrahedron elements at exit points P_e on one face of the tetrahedron element. The time that the particle spends to leave the tetrahedron element from the start point to the exit point is TOF (Time of Flight), which is denoted by $\Delta\tau$. The $\Delta\tau$ and P_e could be computed by the following steps.

Step 1: Calculation of the TOFs.

Assume that one fluid particle could leave a tetrahedron element from any face. Therefore, it is needed to compute the time that the particle spent from start point P_0 to each face, which is denoted by $\Delta\tau_1$, $\Delta\tau_2$, $\Delta\tau_3$, and $\Delta\tau_4$ to Face 1 (face P_2 - P_1 - P_4), Face 2 (face P_2 - P_1 - P_3), Face 3 (face P_3 - P_1 - P_4), and Face 4 (face P_2 - P_3 - P_4), respectively, which is shown in Figure 1b. Taking Case 1 (Section 2.3) as an example, for any given time, the coordinates can be computed; while for given coordinates, the time could be calculated by the Newton–Raphson method. So, when the position coordinate satisfies $\eta = 0$, the $\Delta\tau_1$

could be calculated, which yields Equation (13). In order to compute $\Delta\tau_2$, $\Delta\tau_3$, and $\Delta\tau_4$, the position coordinate should satisfy $\zeta = 0$, $\xi = 0$ and $\xi + \eta + \zeta = 1$, yielding Equations (14)–(16), respectively.

$$0 = \eta = E_{12}e^{r_1\Delta\tau_1} + E_{22}e^{r_2\Delta\tau_1} + E_{32}e^{r_3\Delta\tau_1} + P_{c2} \quad (13)$$

$$0 = \zeta = E_{13}e^{r_1\Delta\tau_2} + E_{23}e^{r_2\Delta\tau_2} + E_{33}e^{r_3\Delta\tau_2} + P_{c3} \quad (14)$$

$$0 = \xi = E_{11}e^{r_1\Delta\tau_3} + E_{21}e^{r_2\Delta\tau_3} + E_{31}e^{r_3\Delta\tau_3} + P_{c1} \quad (15)$$

$$1 = \xi + \eta + \zeta = \begin{aligned} &E_{11}e^{r_1\Delta\tau_4} + E_{21}e^{r_2\Delta\tau_4} + E_{31}e^{r_3\Delta\tau_4} + P_{c1} \\ &+ E_{12}e^{r_1\Delta\tau_4} + E_{22}e^{r_2\Delta\tau_4} + E_{32}e^{r_3\Delta\tau_4} + P_{c2} \\ &+ E_{13}e^{r_1\Delta\tau_4} + E_{23}e^{r_2\Delta\tau_4} + E_{33}e^{r_3\Delta\tau_4} + P_{c3} \end{aligned} \quad (16)$$

Step 2: Determination of the real $\Delta\tau$ from calculated TOFs.

Now, $\Delta\tau_1$, $\Delta\tau_2$, $\Delta\tau_3$, and $\Delta\tau_4$ have been calculated, but these TOFs may be negative or positive. According to the physical meaning of particle flow, the real $\Delta\tau$ should be the minimum value of positive TOFs, which yields $\Delta\tau = \min(\Delta\tau_1, \Delta\tau_2, \Delta\tau_3 \text{ and } \Delta\tau_4)$.

Step 3: Determination of the coordinate of P_e .

Now that the real TOF is determined, the exit coordinate (ξ_e, η_e, ζ_e) could be determined by the following equations.

$$\begin{aligned} \xi_e &= E_{11}e^{r_1\Delta\tau} + E_{21}e^{r_2\Delta\tau} + E_{31}e^{r_3\Delta\tau} + P_{c1} \\ \eta_e &= E_{12}e^{r_1\Delta\tau} + E_{22}e^{r_2\Delta\tau} + E_{32}e^{r_3\Delta\tau} + P_{c2} \\ \zeta_e &= E_{13}e^{r_1\Delta\tau} + E_{23}e^{r_2\Delta\tau} + E_{33}e^{r_3\Delta\tau} + P_{c3} \end{aligned} \quad (17)$$

3. The Method to Trace Streamlines over Tetrahedron Grid Created by FEM

A new method is presented to trace streamlines over the tetrahedron grid system created by FEM, and the flowchart is shown in Figure 2. Based on a given velocity field, the process of tracing one streamline is shown in the steps below:

- (1) Import a tetrahedron mesh system created by FEM and the velocity field.
- (2) Give a start point P_0 in the x - y - z plane (Actual Element); then, find the cell where the start point is located, and obtain these element properties such as the coordinates on each vertex of the cell and the corresponding velocity vector.
- (3) Transform the coordinates and velocity vectors at four vertexes of the tetrahedron element.
- (4) Convert start point properties (such as velocity vector and coordinate) to the Master Element (ξ - η - ζ plane).
- (5) Trace a streamline in the tetrahedron element and determine the exit point P_e^* coordinate and time (TOF) spent from start point to exit point in the Master Element.
- (6) Map the exit point coordinate P_e^* (ξ_e, η_e, ζ_e) in Master Element back to Actual Element as P_e (x_e, y_e, z_e) and accumulate $\Delta\tau$ to construct the TOF coordinate system;
- (7) Check whether the exit point P_e reaches a sink or not. If yes, it means this streamline is traced successfully, and stop tracing this streamline. If does not reach, replace the start point P_0 by using the calculated exit point P_e and repeat Steps 2–7.

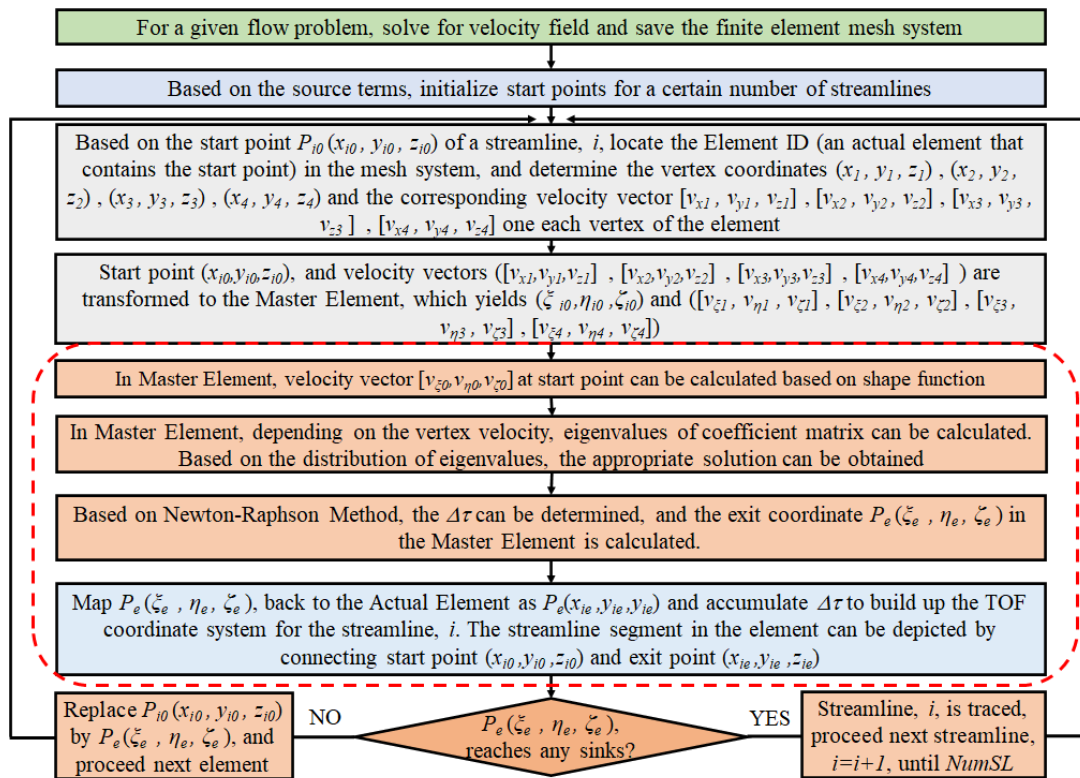


Figure 2. The flowchart of tracing a streamline over the tetrahedron grid system created by Finite Element Method (FEM).

3.1. Tracing Streamline over a Tetrahedron Element

This subsection will elaborate how to trace a streamline over a tetrahedron element in an Actual Element, which includes Step 2 to Step 7.

In **Step 2**, for any given start point $(P_0(x_0, y_0, z_0))$ in the Actual Element, find the tetrahedron element where P_0 is located and obtain the parameters of this tetrahedron element, such as the coordinates of each vertex $(P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3), P_4(x_4, y_4, z_4))$ and velocity vectors on the each vertex $([v_{x1}, v_{y1}, v_{z1}], [v_{x2}, v_{y2}, v_{z2}], [v_{x3}, v_{y3}, v_{z3}], [v_{x4}, v_{y4}, v_{z4}])$.

In **Step 3**, those parameters are used to calculate the start point (ξ_0, η_0, ζ_0) via Equation (5) and the velocity vector on each vertex $([v_{\xi1}, v_{\eta1}, v_{\zeta1}], [v_{\xi2}, v_{\eta2}, v_{\zeta2}], [v_{\xi3}, v_{\eta3}, v_{\zeta3}], [v_{\xi4}, v_{\eta4}, v_{\zeta4}])$ at ξ - η - ζ coordinate of (0,0,0), (1,0,0), (0,1,0), (0,0,1), respectively) in the Master Element via Equation (7).

In **Step 4**, based on the start point coordinate (ξ_0, η_0, ζ_0) and tetrahedron element properties (coordinates and velocity vectors at each vertex), the velocity vector $[v_{\xi0}, v_{\eta0}, v_{\zeta0}]$ at the start point can be calculated by the shape function. Now, the start points properties (coordinate and velocity vectors) and tetrahedron element properties (coordinate and velocity vectors at each vertex of the tetrahedron element) have been converted into Master Element.

In **Step 5**, now it is time to determine the pathway of a streamline over one tetrahedron element, noting that this is only a streamline segment. The equation for illuminating the trajectory of a streamline is presented in Section 2.3. When given a time, t , the position coordinate in the Master Element can be determined. Additionally, the TOF in one tetrahedron element could be calculated according the process expressed in Section 2.4. Then, the exit point coordinate could be calculated by taking the calculated TOF into the corresponding equation (such as Equation (17)).

In **Step 6**, Map the exit point coordinate $P_e^*(\xi_e, \eta_e, \zeta_e)$ in Master Element back from to Actual Element as $P_e(x_e, y_e, z_e)$ via Equation (6).

The procedure of tracing a streamline segment in a tetrahedron element and the relative formulations in Sections 2.3 and 2.4 are demonstrated, which are verified by an example shown

in Figure 3. A tetrahedron is defined in Actual Element with four vertices, which are $P_1(1, 1, 1)$, $P_2(3, 2, 2)$, $P_3(1, 4, 3)$, and $P_4(1, 5, 1)$ whose unit is m, respectively, as well as the corresponding velocity vector at four vertices are also given. When a particle starts from a start point which is located at one face of this tetrahedra element, then it will leave this tetrahedron element at an exit point. The procedure of tracing a streamline segment over a tetrahedron element in the Master Element is shown in Figure 3, which is referred to as Case 1, whose start point $P_0(1.2, 1.1, 1.1)$ is located at Face 1 (Face P_1 - P_2 - P_4) of this tetrahedron element. The corresponding velocity vectors at four vertices of this tetrahedron element are $v_1 = [2, 5, 2]$, $v_2 = [5, 10, 3]$, $v_3 = [2, 10, 4]$, and $v_4 = [5, 10, 5]$, which yields Case 1, whose feature is that Equation (12) has three nonzero and real eigenvalues, as shown in Section 2.3. The velocities unit is m/s.

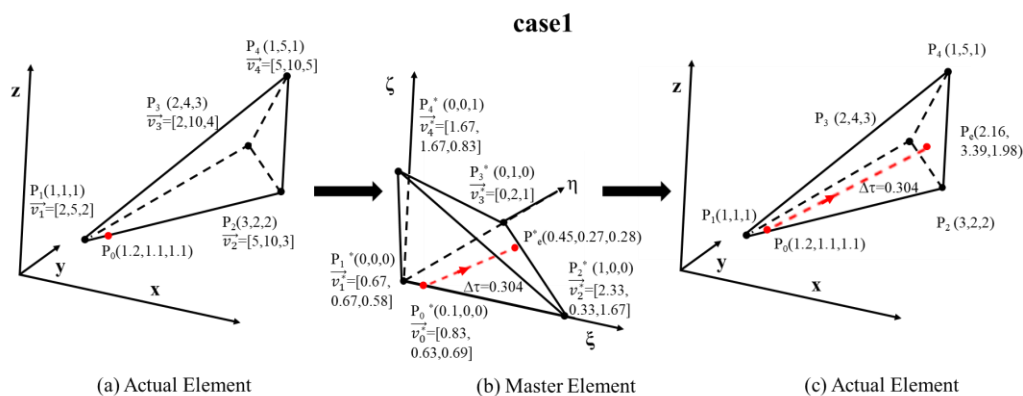


Figure 3. The procedure of tracing a streamline in a tetrahedron element generated by FEM. (a) The coordinates and velocity vectors at each vertex and the start point over a tetrahedron element in Actual Element. (b) The results of coordinates and velocity vectors transformed to Master Element and a streamline trajectory from the start point to the exit point in the Master Element, as well as the Time of Flight (TOF). (c) The streamline trajectory and coordinates of each vertex, start point, and exit point, which mapped back to Actual Element.

The geometrical sketch and parameters of a tetrahedron element in Actual Element (such as the coordinates and velocity vectors) are shown in Figure 3a. Then, coordinates and velocity vectors are transformed into Master Element via Equations (6) and (7), respectively, and the result is shown in Figure 3b. Thereby, the streamline pathway, Time of Flight (TOF) and the exit point coordinate P_e^* are determined by relative formulations in Sections 2.3 and 2.4; the result is also shown in Figure 3b. Finally, the results of the streamline segment (streamline pathway and exit point) could be converted to Actual Element, which is shown in Figure 3c. This example verified the procedure of tracing a streamline segment over a tetrahedron element.

The calculation results by relative formulations in Sections 2.3 and 2.4 are verified by the RK4 method. The RK4 is the so-called fourth-order Runge–Kutta method, which is adopted to verify ODEs, since it is easily to implement during calculation. Equation (10) has nine cases according to the eigenvalue combination caused by the different velocity vectors at each vertex.

Figure 4 shows nine verification cases for Case 1 (three nonzero real eigenvalues shown as Equation (12)), Case 2 (one real and two equal-nonzero real eigenvalues shown as Equation (A1)), Case 3 (a single nonzero real eigenvalue shown as Equation (A2)), Case 4 (one nonzero real and two complex eigenvalues shown as Equation (A3)), Case 5 (one zero real and two nonzero real eigenvalues shown as Equation (A4)), Case 6 (one nonzero real and two zero eigenvalues shown as Equation (A5)), Case 7 (only zero eigenvalues shown as Equation (A6)), Case 8 (one zero and two equal-nonzero real eigenvalues shown as Equation (A7)), Case 9 (two complex and one zero eigenvalues shown as Equation (A8)).

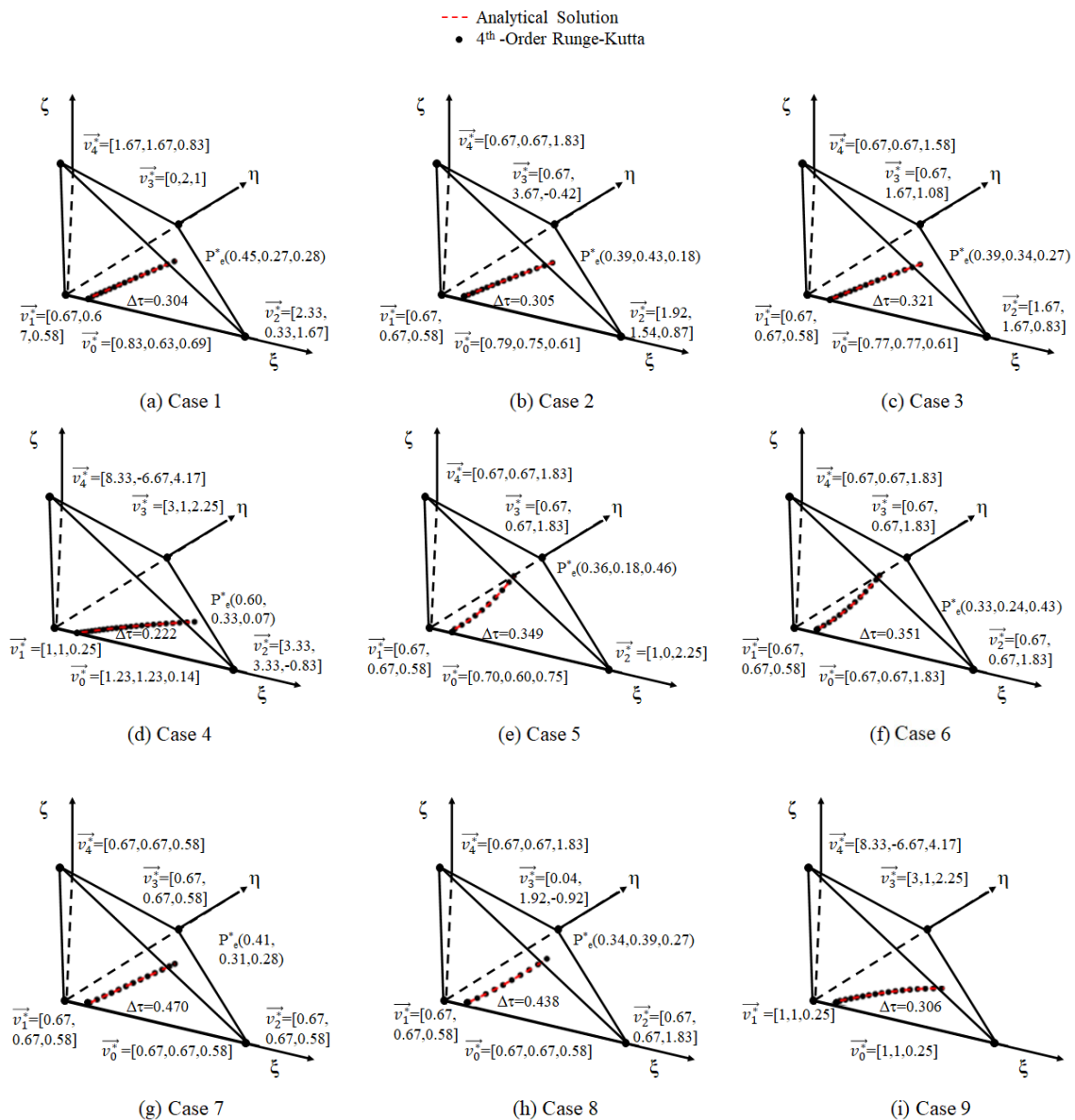


Figure 4. The comparison of this study result (solid lines) and RK4 result (discrete points) for different cases in Master Element. (a–i) represents the result of comparison trajectory between this study and RK4 for Case 1, Case 2, Case 3, Case 4, Case 5, Case 6, Case 7, Case 8, and Case 9, respectively.

The comparison results are shown in Figure 4, including Case 2, where $v_1 = [2,5,2]$, $v_2 = [5.375,10,5]$, $v_3 = [5,10,8]$, and $v_4 = [2,10,2]$; Case 3, where $v_1 = [2,5,2]$, $v_2 = [5,10,5]$, $v_3 = [3,10,4]$ and $v_4 = [2,9,2]$; Case 4, where $v_1 = [3,5,3]$, $v_2 = [10,10,10]$, $v_3 = [7,15,5]$ and $v_4 = [10,5, -5]$; Case 5, where $v_1 = [2,5,2]$, $v_2 = [2,10,1]$, $v_3 = [2,10,2]$, and $v_4 = [2,10,2]$; Case 6, where $v_1 = [2,5,2]$, $v_2 = [2,10,2]$, $v_3 = [2,10,2]$, and $v_4 = [2,10,2]$; Case 7, where $v_1 = [2,5,2]$, $v_2 = [2,5,2]$, $v_3 = [2,5,2]$; and $v_4 = [2,5,2]$, Case 8, where $v_1 = [2,5,2]$, $v_2 = [2,5,2]$, $v_3 = [2,5,3.875]$, and $v_4 = [2,10,2]$, as well as Case 9 where $v_1 = [3,5,3]$, $v_2 = [3,5,3]$, $v_3 = [7,15,5]$, and $v_4 = [10,5, -5]$. The velocities unit is m/s.

All the given velocities are in Actual Element, and all corresponding values are in Master Element, which is denoted as v_1^* , v_2^* , v_3^* , and v_4^* can be found in Figure 4. The streamline for different cases is calculated, and we compared the analytical solution and the 4th-order Runge–Kutta method, and the procedures of using the 4th-order Runge–Kutta method solve partial differential equation (such as Equation (10)), which is shown in Appendix B.

Despite RK4 having numerical errors, if reducing the value of Δt , the results will be convergent to an accurate solution. When $\Delta t = 0.001$ s, all the results of RK4 match perfectly with the result of the analytical solution, which is shown in Section 2.3 in this study. In addition, the TOFs ($\Delta\tau$) calculated by RK4 give equal results in this study (value is shown in Figure 4). Although RK4 can also compute a streamlined pathway, the RK4 is not suited for streamline-based simulation due to its low efficiency and numerical errors.

As is shown in Table 1, the error of TOF and the number of iterations by RK4 is up to large Δt or small Δt . Taking Case 1 as an example, when using small Δt ($\Delta t = 0.0001$ s), the RK4 could calculate an accurate TOF (error of TOF is 0.007%), but it needs to repeat 45645 iterations; while using large Δt ($\Delta t = 0.01$ s), the number of iterations decreased drastically (only 450 iterations), but the error of TOF is 1.42%, which is bigger. Considering the efficiency and the accuracy, 0.001 s is selected to calculate TOF by RK4, which was used to verify the analytical solution. Table 1 implies that although RK4 could determine the trajectory and TOF, there are some problems limiting its applications. (1) The computational accuracy and efficiency depend on the step (Δt) size. A small Δt could determine an accurate TOF and coordinate, which decrease the computational efficiency, while a large Δt could increase the computational efficiency but could not determine an accurate TOF and coordinate. (2) When tracing streamlines in the entire domain, only TOF and exit coordinates are needed to calculate each element, and it is not significant to know the detailed trajectory in one element. However, it requires repeating a large number of iterations to determine the TOF and exit coordinate by the RK4 method compared to only a few iterations in the analytical solution method; the process of determining the exit points and TOF by the analytical solution method is outlined in Section 2.4.

Table 1. The calculation results and error of TOF by analytical solution and RK4 method for nine cases.

Case	TOF Calculated by Analytical Solution	TOF Calculated by RK4								
		$\Delta t = 0.01$ s			$\Delta t = 0.001$ s			$\Delta t = 0.0001$ s		
		TOF (s)	Error (%)	Number of Iterations	TOF (s)	Error (%)	Number of Iterations	TOF (s)	Error (%)	Number of Iterations
Case 1	0.30432	0.3	1.42	450	0.304	0.105	4560	0.3043	0.007	45645
Case 2	0.30525	0.31	−1.556	465	0.305	0.082	4575	0.3051	0.049	45765
Case 3	0.32123	0.32	0.383	480	0.321	0.072	4815	0.3212	0.009	48180
Case 4	0.2214	0.22	0.632	330	0.221	0.181	3330	0.2213	0.045	33195
Case 5	0.34923	0.35	−0.22	375	0.349	0.066	5235	0.3491	0.037	52365
Case 6	0.35111	0.35	0.316	525	0.351	0.031	5265	0.3511	0.003	52665
Case 7	0.46956	0.47	−0.094	705	0.469	0.119	7035	0.4695	0.013	70425
Case 8	0.43823	0.44	−0.404	660	0.438	0.052	6570	0.4382	0.007	64920
Case 9	0.30615	0.31	−1.258	465	0.306	0.049	4590	0.3061	0.016	45915

3.2. Tracing Streamline over the Whole Domain

This subsection demonstrates the process of tracing streamlines in the whole domain and building up a streamline system. After tracing a streamline over a tetrahedron element, it is important to make sure whether the exit point reaches the sink or not. If yes, this streamline is the end. However, if it does not, take the exit point as the start point in the next element and repeat the step of tracing a streamline over a tetrahedron element until the exit point reaches the sink. One streamline is formed by connecting the start point and exit point in each tetrahedron element. The process of tracing streamlines for the whole domain is verified by Case 1, which is shown in Section 4.1.

4. Results and Discussion

Some cases are studied to verify the process of tracing streamlines for the whole domain and illustrate the computational power of the method presented in this paper about tracing streamlines. The purpose of this paper is to propose a general tracing streamline method, so it does not involve any specific physical problems. The following case is based on the Laplace type equation, and the velocity and TOFs are calculated, but not in units. The flow field of the case in this paper is achieved in FEniCS;

the details about FEniCS are shown in <https://fenicsproject.org>. The streamline tracing method in this paper is implemented via the FEniCS framework.

4.1. Case 1

The process of tracing streamlines for the whole domain is verified by a simple fluid flow problem (reference as Case 1), which is shown in Figure 5. Figure 5a is showing a simple fluid flow field of u in a unit cubic domain whose governing equation is Laplace's equation, $\nabla^2 u = 0$, where u , which is scalar, represents the potential field, and v represents the velocity vector $v = -\nabla u$. The boundary conditions are the Neumann boundary condition and Dirichlet boundary condition, where the Dirichlet boundary condition is applied to the left face ($u = 1$ when $x = 0$, y and z equal any values) and the right face ($u = 0$ when $x = 1$, y and z equal any values), while the Neumann boundary condition is applied to the top face ($\partial u / \partial n = 0$ when $y = 1$, x and z equal any values), bottom face ($\partial u / \partial n = 0$ when $y = 0$, x and z equal any values), front face ($\partial u / \partial n = 0$ when $z = 0$, x and y equal any values) and the back face ($\partial u / \partial n = 0$ when $z = 1$, x and y equal any values). So, the flow field (u) in a unit cubic domain is distributed linearly from 1 to 0, which leads to a uniform velocity vector [$v_x = 1$, $v_y = 0$, $v_z = 0$]. Therefore, the streamlines of this flow field are a set of parallel straight lines, such as SL1, SL2, SL3, and SL4 shown in Figure 5b, whose start points are at (0.0, 0.25, 0.25), (0.0, 0.25, 0.75), (0.0, 0.75, 0.25) and (0.0, 0.75, 0.75), respectively.

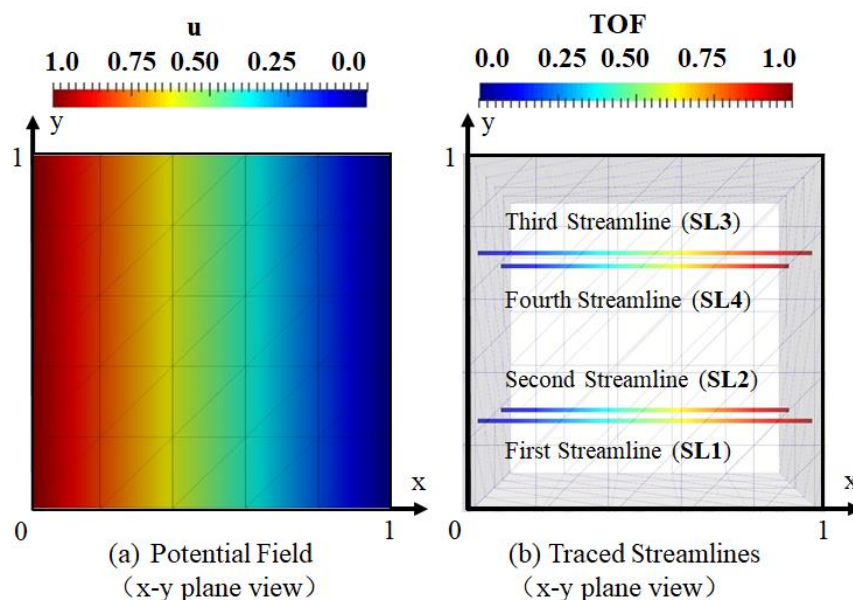


Figure 5. The simple fluid flow problem (Case 1) to verify the process of tracing streamlines for the whole domain where (a) shows the fluid field, u , and (b) shows four traced streamlines whose start points are at (0.0, 0.25, 0.25), (0.0, 0.25, 0.75), (0.0, 0.75, 0.25), and (0.0, 0.75, 0.75), respectively.

As the velocity field is a uniform velocity vector, the TOF of one streamline is linear distributed from 0 at the left face boundary ($x = 0$) to 1 at the right face ($x = 1$), which means the analytical solution of TOF of one streamline could be represented as a line (slope equals one) on the graph of the relationship between TOF and x . Therefore, the TOF of this study could be verified by an exact solution, which is shown in Figure 5b as the solid line. As is shown in Figure 6, the TOFs of four streamlines are in perfect agreement with the analytical solution of TOFs on the plot of TOF and x , which means the new streamline tracing method over the tetrahedron mesh system presented in this paper is accurate.

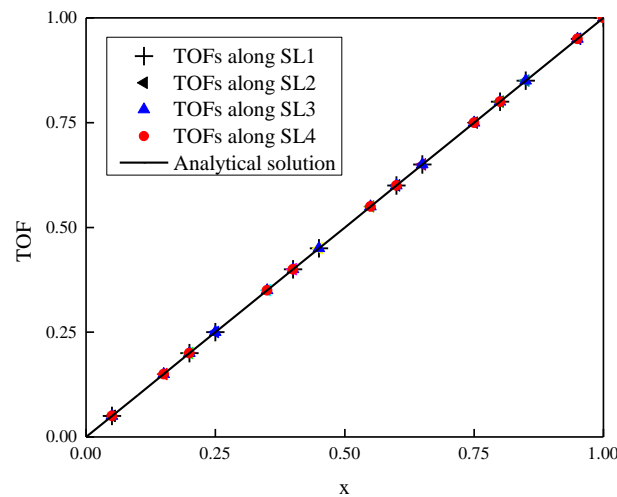


Figure 6. Comparison between analytical TOF shown as solid line and the calculated TOFs along four streamlines (SL1, SL2, SL3, and SL4) shown as discrete points.

4.2. Case 2

Case 2 is used to study the computational power of tracing a streamline in a vertical direction and horizontal direction, respectively. The purpose of this paper is to propose a general tracing streamline method, so it does not involve any specific physical problems.

Case 2 is shown in Figure 7, the calculation domain is a cube with length equaling 1.0, the governing equation is a Laplace equation, whose boundary condition is that the six faces of cube under the no-flow Neumann boundary condition ($\partial u / \partial n = 0$), and the source and sink are under the Dirichlet boundary condition ($u = 1$ and $u = 0$, respectively). The source is noted as a cylinder whose top coordinate is (0.1, 0.1, 1.0), bottom coordinate is (0.1, 0.1, 0.0), and radius is 0.05, while the sink is noted as a cylinder whose top coordinate is (0.9, 0.9, 1.0), bottom coordinate is (0.9, 0.9, 0.0), and radius is 0.05.

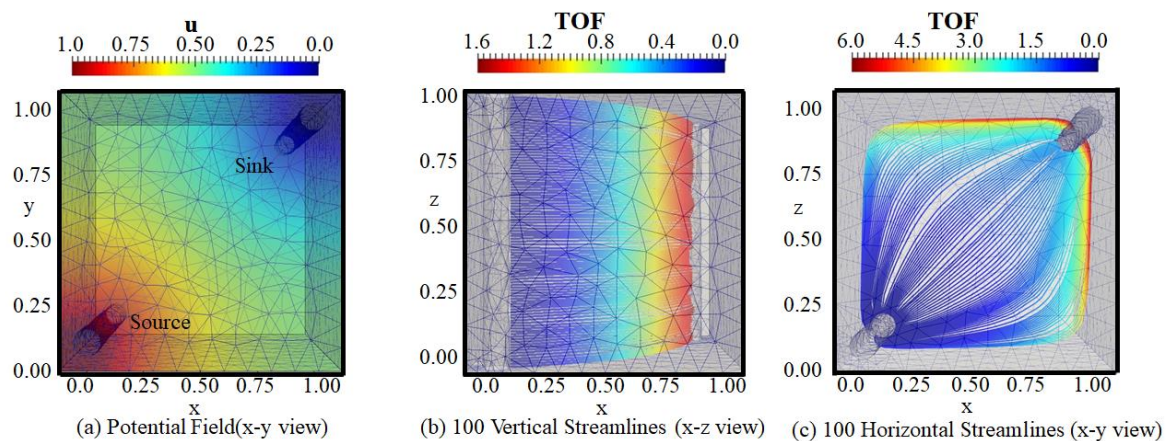


Figure 7. The result of Case 2. (a) The discretization of the calculation domain and the potential field (u), (b,c) show the streamline pattern from the x - y plane view and x - z plane view, respectively.

The discretization of the calculation domain is shown in Figure 7a. In the FEniCS framework, the resolution of a mesh system is determined via specifying a resolution number instead of the average mesh size, such as the '20' in code "mesh = generate_mesh(geometry, 20)". The calculation domain was discretized into 8595 elements. Then, the Laplace equation is solved by FEniCS with boundary conditions in the calculation domain, and it extends the program to compute potential field (u), which is shown in Figure 7a.

Figure 7b presents the result of tracing 100 streamlines whose start points are located at the surface of the source and linearly distributed along the direction of the cylinder centerline. The TOF color map in Figure 7b,c shows the traveling time of a particle spending from the start point to the sink. Near the start point, the color of the streamline is blue, while near the sink, the color of the streamline is red. In Figure 7b, all the streamlines could start from the start point and reach the surface of the sink, which implies that the new method can trace the streamline in the vertical direction successfully.

Figure 7c shows that 100 streamlines were traced in the horizontal direction. The start points of 100 streamlines in Figure 7c are evenly distributed along the horizontal direction at the surface of the source, and all the z coordinates of start points equal 0.5. As shown in Figure 7c, 100 streamlines along the horizontal direction are traced successfully and are evenly distributed on a plane (whose plane equation is $z = 0.5$), which means that the method for tracing a streamline over the tetrahedron grid system presented in this study can trace all the streamlines in the horizontal direction.

According to the streamline tracing results in Figure 7b,c, the method to tracing a streamline over the tetrahedron grid system presented in this study is a robust method.

4.3. Case 3

The method for tracing a streamline over the tetrahedron grid system presented in this study could trace all the streamlines in both the vertical direction and horizontal direction, as implied from Case 2. However, the computational power of the algorithm is not studied when there is an obstacle in the computation domain. So, Case 3 is introduced, where an impermeable sphere is artificially introduced to add computational complexity. The center of the sphere is at (0.5, 0.5, 0.5), and the sphere radius is 0.4. The boundary condition at the surface sphere is the Neumann boundary condition ($\partial u / \partial n = 0$), the boundary conditions are same as those in Case 2. Figure 8a shows the discretization of the calculation domain and the potential field (u), which represents that the fluids flow from the source and bypass the obstacles before finally flowing out from the sink. Figure 8b shows 100 streamlines (10 layers in the vertical direction, 10 streamlines launched in the horizontal direction) are traced successfully, and all the streamlines start from the source, then steer by obstacles and reach the sink.

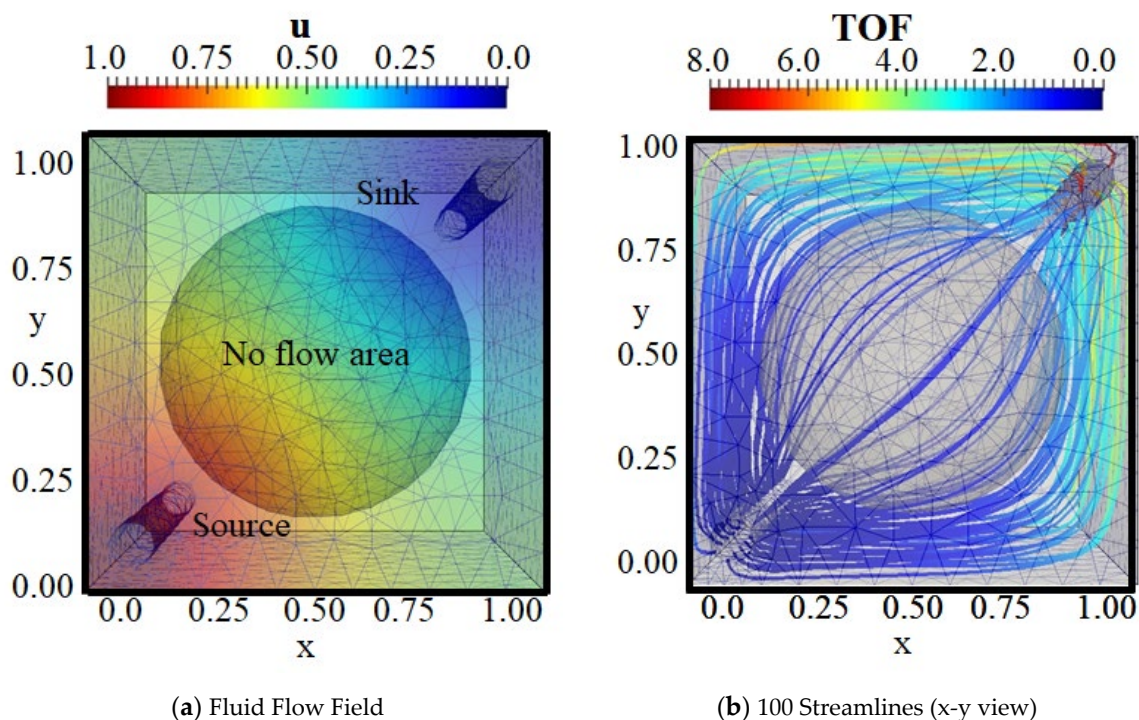


Figure 8. The result of Case 3. (a) The discretization of the calculation domain and the potential field (u), and (b) streamline pattern from the x - y plane view.

5. Conclusions

In this paper, a new method to trace streamlines over unstructured tetrahedron elements based on vertex velocities is presented. The following conclusion are drawn:

- (1) The exit coordinate leaving a tetrahedral element and the Time of Flight (TOF) could be computed accurately and efficiently because the analytical solution that could depict the trajectory in Master Element is deduced.
- (2) Two verification cases are studied, whose results are agreeing to the new method to trace streamlines over unstructured tetrahedron elements based on vertex velocities. One of the verifications verifies the analytical solution for depicting the trajectory in Master Element by comparing it with the results of the RK4 method, while the other one verifies the process of tracing streamlines in the entire domain.
- (3) Two cases with different geometric complexity are studied to demonstrate the computational power of the method presented in this paper. The new method to trace streamlines over unstructured tetrahedron elements is a robust method that can handle a complex computational domain.
- (4) The presented method is implemented in FEniCS, which is a programming framework for Finite Element Method (FEM) that can be easily extend the method to other research fields. If the velocity field and tetrahedron mesh system are obtained during any fluid field solving in FEniCS, the new method can trace the streamlines of this fluid field, which can easily extend the method to other research fields.

Author Contributions: Y.L. contributed to developing the mathematical model, implemented the streamline tracing technique idea, and wrote the manuscript; L.Z. supervised the research and edited the manuscript; Y.F. supported the idea of the streamline tracing technique and built the original code. Y.Z. analyzed the verified the mathematical model and the cases. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded Natural Science Foundation of China (Key Program) (Grant No. 51534006); the National Natural Science Foundation of China (Grant Nos. 51704247, 51874251, 51674044); the PetroChina Innovation Foundation (Grant No. 2018D-5007-0218); the China Scholarship Council (No. 201808510219).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

According to Gregory and II-Hong work (Gregory and II-Hong 1999), Equation (10) has nine solutions depending on the different combinations of eigenvalues. Case 1 is shown as Equation (12), which means A has three nonzero real eigenvalues ($0 \neq r_1 \neq r_2 \neq r_3 \neq 0$), and the other cases are shown as follows.

- (1) Case 2: A has three nonzero real eigenvalues, where two are equal ($0 \neq r_1 \neq r_2 = r_3 \neq 0$)

$$\begin{aligned}
 X(t) &= E_1 e^{r_1 t} + E_2 e^{r_2 t} + E_3 e^{r_3 t} + X_c \\
 E_1 &= \left(\frac{A - r_2 I}{r_2 - r_1} \right)^2 (X_0 - X_c) \\
 E_2 &= \frac{(r_1 I - A)(r_1 I - 2r_2 I + A)}{(r_2 - r_1)^2} (X_0 - X_c) \\
 E_3 &= \frac{(A - r_1 I)(A - r_2 I)}{(r_2 - r_1)} (X_0 - X_c) \\
 X_c &= -BA^{-1}, X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix}
 \end{aligned} \tag{A1}$$

- (2) Case 3: A has one nonzero real eigenvalue ($r_1 = r_2 = r_3 \neq 0$)

$$\begin{aligned} X(t) &= E_1 e^{r_1 t} + E_2 e^{r_2 t} + E_3 e^{r_3 t} + X_c \\ E_1 &= X_0 - X_c \\ E_2 &= (A - r_1 I)(X_0 - X_c) \\ E_3 &= \frac{(A - r_1 I)^2}{2}(X_0 - X_c) \\ X_c &= -BA^{-1}, X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (A2)$$

- (3) Case 4: A has one nonzero real eigenvalue and two complex eigenvalues ($r_1 \neq 0, \mu + \lambda i, \lambda \neq 0$)

$$\begin{aligned} X(t) &= E_1 e^{r_1 t} + E_2 e^{\mu t} \cos(\lambda t) + E_3 e^{\mu t} \sin(\lambda t) + X_c \\ E_1 &= \frac{(A - \mu I)^2 + \lambda^2 I}{(\mu - r_1)^2 + \lambda^2} (X_0 - X_c) \\ E_2 &= \frac{(r_1 I - A)(r_1 I - 2\mu I + A)}{(\mu - r_1)^2 + \lambda^2} (X_0 - X_c) \\ E_3 &= \frac{(A - \mu I)((A - \mu I)(\mu - r_1) + \lambda^2 I)}{(\mu - r_1)^2 + \lambda^2} (X_0 - X_c) \\ X_c &= -BA^{-1}, X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (A3)$$

- (4) Case 5: A has one zero real eigenvalue and two nonzero real eigenvalues ($r_1 = 0, 0 \neq r_2 \neq r_3 \neq 0$)

$$\begin{aligned} X(t) &= E_1 t + E_2 (e^{r_2 t} - 1) + E_3 (e^{r_3 t} - 1) + X_0 \\ E_1 &= \left(\left(\frac{A - r_3 I}{r_2 - r_3} \right) \left(I - \frac{A}{r_2} \right) + \left(\frac{r_2 I - A}{r_2 - r_3} \right) \left(I - \frac{A}{r_3} \right) \right) B \\ E_2 &= \left(\frac{A - r_3 I}{r_2 - r_3} \right) \frac{A}{r_2} \left(X_0 + \frac{B}{r_2} \right) \\ E_3 &= \left(\frac{r_2 I - A}{r_2 - r_3} \right) \frac{A}{r_3} \left(X_0 + \frac{B}{r_3} \right), X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (A4)$$

- (5) Case 6: A has two zero real eigenvalues and one nonzero real eigenvalue ($r_1 = 0, r_2 = 0, r_3 \neq 0$)

$$\begin{aligned} X(t) &= E_1 t + E_2 t^2 + E_3 (e^{r_3 t} - r_3 t - 1) + X_0 \\ E_1 &= AP_0 + B \\ E_2 &= \left(\frac{r_3 I - A}{r_3} \right) \frac{AB}{2} \\ E_3 &= \left(\frac{A}{r_3} \right)^2 \left(X_0 + \frac{B}{r_3} \right), X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (A5)$$

- (6) Case 7: A has only a zero eigenvalue ($r_1 = 0, r_2 = 0, r_3 = 0$)

$$\begin{aligned} X(t) &= E_1 t + E_2 t^2 + E_3 t^3 + X_0 \\ E_1 &= AP_0 + B \\ E_2 &= (AX_0 + B) \frac{A}{2} \\ E_3 &= \frac{A^2 B}{6}, X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (A6)$$

(7) Case 8: A has one zero real eigenvalue and two equal nonzero real eigenvalues ($r_1 = 0$, $r_2 = r_3 \neq 0$)

$$\begin{aligned} X(t) &= E_1 t + E_2 (e^{r_2 t} - 1) + E_3 t e^{r_3 t} + X_0 \\ E_1 &= \left(I + \frac{A}{r_2} \left(\frac{A}{r_2} - 2I \right) \right) B \\ E_2 &= \frac{A}{r_2} \left(2 \left(I - \frac{A}{r_2} \right) \left(X_0 + \frac{B}{r_2} \right) + \frac{A X_0 + B}{r_2} \right) \\ E_3 &= A \left(\frac{A}{r_2} - I \right) \left(X_0 + \frac{B}{r_2} \right), X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (\text{A7})$$

(8) Case 9: A has one zero real eigenvalue and two complex eigenvalues ($r_1 = 0$, $\mu + \lambda i$, $\lambda \neq 0$)

$$\begin{aligned} X(t) &= E_1 e^{r_1 t} + E_2 e^{\mu t} \cos(\lambda t) + E_3 e^{\mu t} \sin(\lambda t) + X_0 \\ E_1 &= \left(I - A \frac{2\mu I - A}{\mu^2 + \lambda^2} \right) B \\ E_2 &= \frac{A}{\mu^2 + \lambda^2} ((2\mu I - A)X_0 + G_1 B) \\ E_3 &= \frac{A}{\lambda(\mu^2 + \lambda^2)} (G_2 X_0 + G_3 B) \\ G_1 &= \frac{3\mu^2 I - \lambda^2 I - 2\mu A}{\mu^2 + \lambda^2}, G_2 = (\mu A - (\mu^2 - \lambda^2)I) \\ G_3 &= \frac{(\mu^2 - \lambda^2)A - \mu(\mu^2 - 3\lambda^2)I}{\mu^2 + \lambda^2}, X_0 = \begin{pmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{pmatrix} \end{aligned} \quad (\text{A8})$$

Appendix B

The procedures of using the 4th-Order Runge–Kutta method to solve a partial differential equation (such as Equation (10)) are summarized as follows.

$$\begin{aligned} K_0 &= h(a_1 + b_1 \xi_n + c_1 \eta_n + d_1 \zeta_n) \\ L_0 &= h(a_2 + b_2 \xi_n + c_2 \eta_n + d_2 \zeta_n) \\ M_0 &= h(a_3 + b_3 \xi_n + c_3 \eta_n + d_3 \zeta_n) \end{aligned} \quad (\text{A9})$$

$$\begin{aligned} K_1 &= h \left[a_1 + b_1 \left(\xi_n + \frac{1}{2} K_0 \right) + c_1 \left(\eta_n + \frac{1}{2} L_0 \right) + d_1 \left(\zeta_n + \frac{1}{2} M_0 \right) \right] \\ L_1 &= h \left[a_2 + b_2 \left(\xi_n + \frac{1}{2} K_0 \right) + c_2 \left(\eta_n + \frac{1}{2} L_0 \right) + d_2 \left(\zeta_n + \frac{1}{2} M_0 \right) \right] \\ M_1 &= h \left[a_3 + b_3 \left(\xi_n + \frac{1}{2} K_0 \right) + c_3 \left(\eta_n + \frac{1}{2} L_0 \right) + d_3 \left(\zeta_n + \frac{1}{2} M_0 \right) \right] \end{aligned} \quad (\text{A10})$$

$$\begin{aligned} K_2 &= h \left[a_1 + b_1 \left(\xi_n + \frac{1}{2} K_1 \right) + c_1 \left(\eta_n + \frac{1}{2} L_1 \right) + d_1 \left(\zeta_n + \frac{1}{2} M_1 \right) \right] \\ L_2 &= h \left[a_2 + b_2 \left(\xi_n + \frac{1}{2} K_1 \right) + c_2 \left(\eta_n + \frac{1}{2} L_1 \right) + d_2 \left(\zeta_n + \frac{1}{2} M_1 \right) \right] \\ M_2 &= h \left[a_3 + b_3 \left(\xi_n + \frac{1}{2} K_1 \right) + c_3 \left(\eta_n + \frac{1}{2} L_1 \right) + d_3 \left(\zeta_n + \frac{1}{2} M_1 \right) \right] \end{aligned} \quad (\text{A11})$$

$$\begin{aligned} K_3 &= h[a_1 + b_1(\xi_n + K_2) + c_1(\eta_n + L_2) + d_1(\zeta_n + M_2)] \\ L_3 &= h[a_2 + b_2(\xi_n + K_2) + c_2(\eta_n + L_2) + d_2(\zeta_n + M_2)] \\ M_3 &= h[a_3 + b_3(\xi_n + K_2) + c_3(\eta_n + L_2) + d_3(\zeta_n + M_2)] \end{aligned} \quad (\text{A12})$$

$$\begin{aligned} \xi_n &= \xi_{n+1} + \frac{1}{6}(K_0 + 2K_1 + 2K_2 + K_3) \\ \eta_n &= \eta_{n+1} + \frac{1}{6}(L_0 + 2L_1 + 2L_2 + L_3) \\ \zeta_n &= \zeta_{n+1} + \frac{1}{6}(M_0 + 2M_1 + 2M_2 + M_3) \end{aligned} \quad (\text{A13})$$

where h is step size (small incremental traveling time, $\Delta\tau$, as used in this study), ξ_n , η_n , and ζ_n represent the coordinate in the master element at current iterative step, and ξ_{n+1} , η_{n+1} , and ζ_{n+1} are for the next iterative step. Via solving Equations (A9)–(A13) sequentially and repeating the iterative procedure, the trajectory can be depicted in the master element, and the exit coordinate as well as the corresponding traveling time (TOF) can be determined accordingly.

References

1. Gautier, Y.; Noetinger, B.; Roggero, F. History Matching Using a Streamline-Based Approach and Gradual Deformation. *SPE J.* **2004**, *9*, 88–101. [\[CrossRef\]](#)
2. Singh, A.P.; Knabe, S.P.; Maucec, M. History Matching Using Streamline Trajectories. In Proceedings of the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE, 10–13 November 2013; Society of Petroleum Engineers (SPE): London, UK, 2014; pp. 10–13.
3. Ramirez, W.F. *Application of Optimal Control Theory to Enhanced Recovery*; Development of Petroleum Science series; Elsevier Science: New York, NY, USA, 1987; Volume 21.
4. Siavashi, M.; Tehrani, M.R.; Nakhaee, A. Efficient Particle Swarm Optimization of Well Placement to Enhance Oil Recovery Using a Novel Streamline-Based Objective Function. *J. Energy Resour. Technol.* **2016**, *138*, 052903. [\[CrossRef\]](#)
5. Zhu, Z.; Gerritsen, M.; Thiele, M. Thermal Streamline Simulation for Hot Waterflooding. *SPE Reserv. Eval. Eng.* **2010**, *13*, 372–382. [\[CrossRef\]](#)
6. Wen, T.; Thiele, M.R.; Ciaurri, D.E.; Aziz, K.; Ye, Y. Waterflood management using two-stage optimization with streamline simulation. *Comput. Geosci.* **2014**, *18*, 483–504. [\[CrossRef\]](#)
7. Batycky, R.P. A Three-Dimensional Two-Phase Field Scale Streamline Simulator. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 1997.
8. Alsofi, A.M.; Blunt, M.J. Streamline-Based Simulation of non-Newtonian Poly Flooding. *SPE J.* **2010**, *15*, 895–905. [\[CrossRef\]](#)
9. Huang, W.; di Donato, G.; Blunt, M.J. Comparison of streamline-based and grid-based dual porosity simulation. *J. Pet. Sci. Eng.* **2004**, *43*, 129–137. [\[CrossRef\]](#)
10. Park, C.; Kang, J.M.; Jung, Y.; Kim, S. Streamline-based simulation to investigate interwell connectivity and tracer transport in 3D discrete fractured reservoir. In Proceedings of the SPE Europe/EAGE Annual Conference and Exhibition, Vienna, Austria, 12–15 June 2006.
11. Atchley, A.L.; Maxwell, R.M.; Navarre-Sitchler, A.K. Using streamlines to simulate stochastic reactive transport in heterogeneous aquifers: Kinetic metal release and transport in CO₂ impacted drinking water aquifers. *Adv. Water Resour.* **2013**, *52*, 93–106. [\[CrossRef\]](#)
12. Qi, R.; la Force, T.; Blunt, M.J. A three-phase four-component streamline-based simulator to study carbon dioxide storage. *Comput. Geosci.* **2009**, *13*, 493–509. [\[CrossRef\]](#)
13. Crane, M.J.; Blunt, M.J. Streamline-based simulation of solute transport. *Water Resour. Res.* **1999**, *35*, 3061–3078. [\[CrossRef\]](#)
14. Maxwell, R.M.; Welty, C.; Tompson, A.F.B. Streamline-based simulation of virus transport resulting from long term artificial recharge in a heterogeneous aquifer. *Adv. Water Resour.* **2003**, *26*, 1075–1096. [\[CrossRef\]](#)
15. Di Donato, G.; Blunt, M.J. Streamline-based dual-porosity simulation of reactive transport and flow in fractured reservoirs. *Water Resour. Res.* **2004**, *40*, 1–12. [\[CrossRef\]](#)
16. Obi, E.-O.; Blunt, M.J. Streamline-based simulation of advective–dispersive solute transport. *Adv. Water Resour.* **2004**, *27*, 913–924. [\[CrossRef\]](#)
17. Herrera, P.A.; Valocchi, A.J.; Beckie, R.D. A multidimensional streamline-based method to simulate reactive solute transport in heterogeneous porous media. *Adv. Water Resour.* **2010**, *33*, 711–727. [\[CrossRef\]](#)
18. Batycky, R.P.; Thiele, M.R. Technology Update: Mature Flood Surveillance Using Streamlines. *J. Pet. Technol.* **2016**, *68*, 22–25. [\[CrossRef\]](#)
19. Zefreh, M.G.; Nilsen, H.M.; Lie, K.; Raynaud, X.; Doster, F. Streamline simulation of a reactive advective flow with discontinuous flux function. *Comput. Geosci.* **2018**, *23*, 255–271. [\[CrossRef\]](#)
20. Caudle, B.H. *Fundamentals of Reservoir Engineering*; Society of Petroleum: Richardson, TX, USA, 1967.
21. Collins, R.E. Japanese research in flow of fluids through porous materials and filtration. *Trans. Am. Geophys. Union* **1964**, *45*, 151–164. [\[CrossRef\]](#)
22. Lin, J.K. An Image well Method for Bounding Arbitrary Reservoir Shapes in the Streamline Model. Ph.D. Thesis, University of Texas at Austin, Austin, TX, USA, 1972.
23. Thiele, M.R. Modeling Multiphase Flow in Heterogeneous Media Using Streamtubes. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 1994.

24. Datta-Gupta, A.; King, M.J. A semi analytical approach to tracer flow modeling in heterogeneous permeable media. *Adv. Water Res.* **1995**, *18*, 9–29. [[CrossRef](#)]
25. Thiele, M.R. Streamline simulation. In *6th International Forum on Reservoir Simulation*; Schloss Fuschl: Salzburg, Austria, 2001.
26. Datta-Gupta, A.; King, M.J. *Streamline Simulation: Theory and Practice*; Society of Petroleum Engineering: Houston, TX, USA, 2007.
27. Hægland, H.; Dahle, H.; Eigestad, G.; Lie, K.-A.; Aavatsmark, I. Improved streamlines and time-of-flight for streamline simulation on irregular grids. *Adv. Water Resour.* **2007**, *30*, 1027–1045. [[CrossRef](#)]
28. Numbere, D.T. A General Streamline Modeling Technique for Homogeneous Porous with Application to Streamflood Prediction. Ph.D. Thesis, University of Oklahoma, Norman, OK, USA, 1982.
29. Pollock, D.W. Semianalytical Computation of Path Lines for Finite-Difference Models. *Ground Water* **1988**, *26*, 743–750. [[CrossRef](#)]
30. Prévost, M.; Edwards, M.G.; Blunt, M.J. Streamline Tracing on Curvilinear Structured and Unstructured Grids. *SPE J.* **2002**, *7*, 139–148. [[CrossRef](#)]
31. Juanes, R.; Matringe, S.F. Unified formulation for highorder streamline tracing on two-dimensional unstructured grids. *J. Sci. Comput.* **2008**, *38*, 50–73. [[CrossRef](#)]
32. Zhang, Y.; King, M.J.; Datta-Gupta, A. Robust streamline tracing using inter-cell fluxes in locally refined and unstructured grids. *Water Resour. Res.* **2012**, *48*, 1–19. [[CrossRef](#)]
33. Siavashi, M.; Blunt, M.J.; Raisee, M.; Pourafshary, P. Three-dimensional streamline-based simulation of non-isothermal two-phase flow in heterogeneous porous media. *Comput. Fluids* **2014**, *103*, 116–131. [[CrossRef](#)]
34. Zuo, L.; Lim, J.; Chen, R.; King, M. Efficient Calculation of Flux Conservative Streamline Trajectories on Complex and Unstructured Grids. In Proceedings of the 78th EAGE Conference and Exhibition 2016, Vienna, Austria, 30 May–2 June 2016; EAGE Publications: Vienna, Austria.
35. Vasco, D.W.; Pride, S.R.; Zahasky, C.; Benson, S.M. Calculating Trajectories Associated with Solute Transport in a Heterogeneous Medium. *Water Resour. Res.* **2018**, *54*, 6890–6908. [[CrossRef](#)]
36. Wang, B.; Feng, Y.; Du, J.; Wang, Y.; Wang, S.; Yang, R. An Embedded Grid-Free Approach for Near-Wellbore Streamline Simulation. *SPE J.* **2018**, *23*, 567–588. [[CrossRef](#)]
37. Cordes, C.; Kinzelbach, W. Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite element. *Water Resour. Res.* **1992**, *28*, 2903–2911. [[CrossRef](#)]
38. Ham, D.A.; Pietrzak, J.; Stelling, G.S. A streamline tracking algorithm for semi-Lagrangian advection schemes based on the analytic integration of the velocity field. *J. Comput. Appl. Math.* **2006**, *192*, 168–174. [[CrossRef](#)]
39. Matringe, S.F.; Juanes, R.; Tchilepi, H.A. Streamline Tracing on General Triangular or Quadrilateral Grids. *SPE J.* **2007**, *12*, 217–233. [[CrossRef](#)]
40. Park, C.; Kang, J.M. Characterization of Tracer Transport in Heterogeneous Fractured Reservoir by a Streamline Approach on Unstructured Grids. *Energy Sources* **2008**, *30*, 856–871. [[CrossRef](#)]
41. Klausen, R.A.; Rasmussen, A.F.; Stephansen, A.F. Velocity interpolation and streamline tracing on irregular geometries. *Comput. Geosci.* **2011**, *16*, 261–276. [[CrossRef](#)]
42. Painter, S.L.; Gable, C.W.; Kelkar, S. Pathline tracing on fully unstructured control-volume grids. *Comput. Geosci.* **2012**, *16*, 1125–1134. [[CrossRef](#)]
43. Hao, J.; Wang, W.; Duo, Z. An efficient algorithm for streamline visualization in unstructured steady vector field. In Proceedings of the 2nd International Conference on Computer Science and Application Engineering, Hohhot, China, 22–24 October 2018.
44. Yin, F.; Erxiu, S.; Yi, L.; Bin, W.; Liehui, Z.; Yulong, Z. Implementation of streamline simulation based on finite element method in FEniCS. *Comput. Geosci.* **2019**, *24*, 1–5.
45. Bonduà, S.; Bortolotti, V. A tool for pathline creation using TOUGH simulation results and fully unstructured 3D Voronoi grids. *Environ. Model. Softw.* **2019**, *115*, 63–75. [[CrossRef](#)]
46. Craig, J.; Ramadhan, M.; Muffels, C. A Particle Tracking Algorithm for Arbitrary Unstructured Grids. *Ground Water* **2019**, *58*, 19–26. [[CrossRef](#)] [[PubMed](#)]

47. Batista, D. Mesh-independent streamline tracing. *J. Comput. Phys.* **2020**, *401*, 108967. [[CrossRef](#)]
48. Gregory, M.N.; II-Hong, J. Tools for computing tangent curves for linearly baring vector fields over tetrahedral domains. *IEEE Trans. Vis. Comput. Graph.* **1999**, *5*, 360–372.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).