IMPROVED STREAMLINES AND TIME-OF-FLIGHT FOR STREAMLINE SIMULATION ON IRREGULAR GRIDS

H. HÆGLAND, H. K. DAHLE, G. T. EIGESTAD, K.-A. LIE, AND I. AAVATSMARK

Abstract. Streamline methods have shown to be effective for reservoir simulation. For a regular grid, it is common to use the semi-analytical Pollock's method to obtain streamlines and time-of-flight coordinates (TOF). The usual way of handling irregular grids is by trilinear transformation of each grid cell to a unit cube together with a linear flux interpolation scaled by the Jacobian. The flux interpolation allows for fast integration of streamlines, but is inaccurate even for uniform flow. To improve the tracing accuracy, we introduce a new interpolation method, which we call *corner velocity interpolation*. Instead of interpolating the velocity field based on discrete fluxes at cell edges, the new method interpolates directly from reconstructed point velocities given at the corner points in the grid. This allows for reproduction of uniform flow, and eliminates the influence of cell geometries on the velocity field. Using several numerical examples, we demonstrate that the new method is more accurate than the standard tracing methods.

1. Introduction

Streamlines, pathlines, and streaklines are convenient tools for describing and visualizing flow given by an external velocity field $\boldsymbol{q} = (q_x, q_y, q_z)$. *Streamlines* are a family of curves $\boldsymbol{s}(\tau)$ that are instantaneously tangent to the velocity vector $\boldsymbol{q}$ at every point

$$\frac{d\boldsymbol{s}}{d\tau} = \boldsymbol{q}.$$

Streamlines can be traced for any vector field, although the most common is that $\boldsymbol{q}$ represents a velocity obtained from the solution of a set of flow equations. For incompressible flow, streamlines defined at a single instant do not intersect and cannot begin or end inside the fluid. *Streamtubes* are regions bounded by streamlines. Because streamlines are tangent to the velocity field, fluid that is inside a streamtube must remain forever within the same streamtube.

A *pathline* $\boldsymbol{x}(t)$ is the trajectory traced out by an imaginary massless particle following the flow of the fluid from a given starting point,

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{q}(\boldsymbol{x}, t), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{1}$$

A *streakline* is the locus at a given instance of the positions of all fluid particles that have gone through a fixed spatial point in the past. In steady

Date: September 18, 2006.

Key words and phrases. Streamline tracing, Time-of-flight, Irregular grids.

flow streamlines, streaklines, and pathlines coincide; in unsteady flow they can be different.

The integration of (1) to obtain particle paths and/or travel times, is known as *particle tracking*, for which there exists a rich literature. The particle tracking literature is primarily concerned with problems where the velocity field is only known at a finite set of points, either measured or calculated from a flow model, and interpolation is needed to integrate pathlines. In computational fluid dynamics, particle tracking has been used for visualization [33, 35, 51, 56, 57]. Velocity interpolation in control-volume mixed finite-element methods is a related subject to particle tracking and has been considered in [43]. Within groundwater flow simulation, particle tracking is used to model contaminant transport [6, 15, 45, 50, 52, 53]. In visualization, the integration of (1) is usually done numerically using a Runge–Kutta type solver, whereas in groundwater flow, semi-analytical integration is the most common.

In the following we regard *streamline tracing* as a subset of particle tracking, since streamlines may be computed by particle tracking if we introduce the streamline parameter τ as an artificial time variable for which the instantaneous velocity field $\boldsymbol{q}(\boldsymbol{x}, t)$ is steady. In this paper we consider streamline tracing in the context of streamline simulation of flow in hydrocarbon reservoirs [3, 7, 32]. In this case, the fluid velocity $\boldsymbol{q}$ is typically given as the numerical solution of a set of flow equations for $\boldsymbol{q}$ and the fluid pressure p of the form

$$cp_t + \nabla \cdot \boldsymbol{q} = b, \qquad \boldsymbol{q} = -\boldsymbol{a}(\boldsymbol{x})\nabla p.$$

The two equations are commonly referred to as the pressure equation and Darcy's law, respectively. How the corresponding discrete velocity approximation is defined, depends on the numerical method:

- For finite-difference methods, the pressure is usually computed at cell centers, and fluxes can be obtained at cell edges by application of a discrete form of Darcy's law [63].
- For finite-element methods, the numerical solution gives a continuously defined pressure approximation given as the sum of the basis functions for all elements weighted by the corresponding node values. Although a continuously defined velocity can be obtained from Darcy's law, a better strategy is given in [15, 19], where continuous fluxes are obtained at cell edges.
- Mixed finite-element methods *solve* for velocity and pressure simultaneously, resulting in a more accurate velocity field than for finite differences and standard finite elements. The continuously defined velocity is given by the degrees of freedom at the edges and the corresponding basis functions [19, 31]; see also [30, 41, 42].
- Finite-volume methods include multi-point flux approximations [1, 20] and control-volume finite-element methods [10, 62]. In these methods fluxes are computed at cell edges.

In other words, a continuously defined velocity field is obtained only for the mixed finite-element method. For the other methods one must use an interpolation scheme to determine the velocity from the discrete fluxes at the cell edges.

In reservoir simulation and groundwater flow, the predominant way of computing streamlines is by use of a semi-analytical technique. In semi-analytical methods [24, 38, 47, 52], the interpolation of the velocity is simple enough that analytical integration is possible within each grid cell. As an example, let us consider the popular method of Pollock [47]. Given an entry point of a streamline into a grid cell, Pollock's method starts by mapping the grid cell onto the unit square (or unit cube in 3D). Each component of the velocity field is then approximated in reference space by a linear function, in which case the streamline path in each direction is given as an exponential function of the travel time. To trace the streamline, Pollock's method determines the travel time through the grid block as the minimum time to exit in each spatial direction, which is given by a logarithmic expression. Then the travel time is used to compute the exit point and the exit point is mapped back into physical space to give the entry point into the next cell, and so on. In groundwater flow and visualization, more complicated interpolation schemes have been used, where numerical integration is needed, using Euler's method or higher-order Runge-Kutta methods [6, 12, 45, 46, 53].

Tracing of streamlines for use in flow simulations has been investigated in [28, 30, 32, 39, 40, 41, 42, 48, 49]. In the current paper we focus on irregular grids in three spatial dimensions consisting of hexahedral grid cells with curved surfaces. Streamline tracing may then be performed by a method due to Prévost et al. [49]. This method is a simplification of work done by Cordes and Kinzelbach [15], where Pollock's method is extended to irregular grids. Each grid cell in physical space $\mathcal{P}$ is transformed to a unit cube in a reference space $\mathcal{R}$ using a standard isoparametric trilinear transformation [14]. Next, the velocity in $\mathcal{R}$ is approximated by a linear flux interpolation scaled by an approximation of the Jacobian determinant of the transformation. Finally, the streamline segments passing through each cell can be integrated in $\mathcal{R}$ and mapped back to $\mathcal{P}$. Henceforth, we will call this method *standard flux mapping*, SFM.

As we shall see, the standard flux mapping cannot reproduce a uniform flow field on irregular hexahedrons. Uniform flow or almost uniform flow are important cases to consider, since such flow patterns are likely to occur in large parts of a hydrocarbon reservoir. For example, if the driving force for the flow in the reservoir is gravity, and the permeability of the medium is almost homogeneous, the flow locally is almost uniform in the z-direction of the reservoir. Reproduction of uniform flow is important if the grid cells are small compared to the variation of the velocity. Also, the failure to reproduce uniform flow will produce errors in the interpolated velocity for the SFM even for nonuniform flow. The error will increase with the irregularity of the grid cells.

To handle the difficulty associated with the standard flux mapping, we consider an alternative method, which we call *corner velocity interpolation* (CVI). Instead of interpolating the velocity field based on discrete fluxes at cell edges, we interpolate directly from point velocities given at the corner points in the grid. This allows for reproduction of uniform flow, and eliminates the influence of the cell geometry on the velocity field. In streamline simulation, we usually only know fluxes, so we also present a method for reconstructing the corner-point velocities in each grid cell from fluxes.

A radically different strategy was recently introduced by Matringe et al. [41, 42] and Juanes and Matringe [30] for handling irregular grids consisting of triangles and quadrilaterals in 2D. They propose both low and high-order tracing methods based on the mathematical framework of mixed finite-element methods and the associated velocity spaces.

The rest of this paper is organized as follows: First we give a short introduction to streamline simulation. Then Section 3 describes the standard method for streamline tracing on irregular grids. We start with a brief review of Pollock's method [47] for streamline tracing on a Cartesian grid. Then a velocity transformation from physical to reference space is described. Considering this velocity transformation we discuss the methods of Cordes and Kinzelbach [15] and Prévost et al. [49] for streamline tracing on irregular grids. A recent improvement of this standard flux mapping due to Jimenez et al. [28] is also discussed. The latter method is here denoted by *extended flux mapping* (EFM). Finally, the extension of these methods to 3D will motivate the corner velocity interpolation method (CVI).

In Section 4, the CVI method is described first for the 2D case and then extended to 3D. We use bilinear or trilinear interpolation of the velocities at the corners of the cell [11, 57]; an important part of the method is the reconstruction of the corner velocities from the given fluxes.

Finally, in Section 5 the three methods (SFM, EFM, and CVI) are compared by numerical experiments for different grids and flow fields in 3D. We consider uniformly and nonuniformly refined grids, including random hexahedral, pyramidal, and a simplistic real field grid. The flow fields may be analytical solutions of the pressure equation, including uniform, combined uniform and nonuniform, and pure nonuniform flow; or, realistic flow fields where analytical solutions are not available. Further, the CVI method is also tested for use with half-edge fluxes in 2D.

2. Background: Streamline Simulation

Multiphase flow in porous media is usually modeled by a coupled set of differential equations. Using the so-called fractional flow formulation, the flow of e.g., two phases can be described by a parabolic equation for the fluid pressure p (neglecting for simplicity gravity and capillary forces)

$$c_t \partial_t p + \nabla \cdot \boldsymbol{q} = b_p, \tag{2}$$

where $\boldsymbol{q}$ is the total velocity (sum of phase velocities), c_t is the total compressibility, and b_p is a source term. Equation (2) is linked to a transport equation for the fluid saturation S

$$\phi \partial_t S + \nabla \cdot \big(\boldsymbol{q} f(S, \boldsymbol{x})\big) = b_s, \tag{3}$$

through Darcy's equation for the velocity,

$$\boldsymbol{q} = -\lambda(S, \boldsymbol{x}) \nabla p. \tag{4}$$

Here, ϕ, λ, f, and b_s denote porosity, total mobility, fractional flow function, and source terms, respectively.

The basis for any streamline simulation method is a sequential splitting of the coupled pressure and saturation equations, in which one first fixes the saturation and solves the pressure equation (2) and Darcy's law (4). The pressure and velocity fields are then used as parameters while advancing the transport equation (3) a given time step. Finally, the new saturation field is used as input parameter for a new pressure solution step, and so on.

In reservoir simulation, the streamline parameter τ is called time-of-flight, since it can be interpreted as the travel time of a neutral particle along the streamline. Together with the bi-stream functions ψ and χ, for which $\rho\boldsymbol{q} = \nabla\psi \times \nabla\chi$, the time-of-flight τ forms an alternative set of coordinates for three-dimensional space [4, 13, 32]. Here, the effective density ρ reduces to $\rho \equiv 1$ for incompressible flows, see [13]. The Jacobian of the transformation from physical coordinates (x, y, z) to time-of-flight coordinates (τ, ψ, χ) simply equals ϕ. Using this, and the fact that $\boldsymbol{q}$ is orthogonal to $\nabla\psi$ and $\nabla\chi$, allows us to simplify the directional gradient along $\boldsymbol{q}$ as follows:

$$\boldsymbol{q} \cdot \nabla = \phi \frac{\partial}{\partial \tau}.$$

This operator identity is a key point in any streamline method, allowing the multidimensional transport equation (3) to be transformed to a family of one-dimensional transport equations along streamlines (which are straight lines in (τ, ψ, χ) space),

$$\partial_t S + \partial_\tau f(S) = b_s - f(S)\nabla \cdot \boldsymbol{q}. \tag{5}$$

The last term on the right-hand side accounts for compression or expansion of fluids in the case of compressible flows. For incompressible flows, $\nabla \cdot \boldsymbol{q} = 0$ outside wells. Solving the family of one-dimensional problems (5) on a representative set of streamlines is often much faster than solving (3) over a grid in physical space.

Streamline simulation has grown in popularity in the last years due to its ability to deliver fast and accurate simulation of large reservoir models using simplified flow physics [32, 58]. However, current streamline simulators are also capable of including gravity and capillary effects by the means of operator splitting [8, 22, 23] and simulating complex flow physics like miscibility [29, 61], three-phase [26, 37] and compositional flow [16, 59], and dual-porosity models [17, 60].

3. Streamline Tracing on Irregular Grids

In this section we will describe the standard method that is used in current commercial streamline codes for tracing streamlines on irregular grids. The method relies on a trilinear mapping from physical space to a reference space, linear interpolation of each velocity component, and analytical solution of the streamline equation (1) within each grid cell. We will then show that the method, and its recent extension due to Jimenez et al. [28], are unable to correctly reproduce uniform flow on arbitrary irregular grids in three spatial dimensions.

3.1. **Introduction: Tracing on Cartesian Grids.** To motivate the description of streamline tracing on irregular grids, we start by discussing the basic version on Cartesian grids, which is commonly referred to as Pollock's

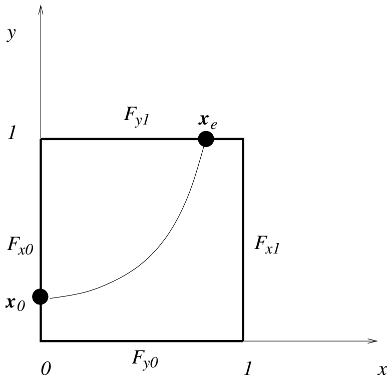

FIGURE 1. Streamline tracing on a unit square.

method in the literature. As we saw in the introduction, Pollock's method builds a streamline as a series of (small) line segments that each cross a grid cell in physical space. The segments are constructed such that the exit point of the streamline in one cell is the entrance point in the next cell. For the development in this paper, it is sufficient to consider the method in the unit square (or unit cube in 3D).

Pollock's Method. The method will be presented for 2D; the extension to 3D should be obvious. Linear interpolation of the edge fluxes is then used to define a velocity field (see Figure 1)

$$\boldsymbol{q}^{\mathrm{I}}(x,y) \equiv \begin{bmatrix} F_{x0}(1-x)+F_{x1}x \\ F_{y0}(1-y)+F_{y1}y \end{bmatrix}, \qquad 0 \le x \le 1, \quad 0 \le y \le 1. \tag{6}$$

Here the superscript I refers to the fact that the velocity field is interpolated based on fluxes given at the edges, as shown in Figure 1.

Having defined a velocity field, the streamline $\boldsymbol{s}(t) = [x(t), y(t)]$ is found by integrating the system of ODEs in (1):

$$\begin{cases} \dfrac{dx}{dt} = q_x^{\mathrm{I}}(x), & x(0) = x_0, \\ \dfrac{dy}{dt} = q_y^{\mathrm{I}}(y), & y(0) = y_0, \end{cases} \tag{7}$$

where q_x^{I} and q_y^{I} are the x- and y-components of $\boldsymbol{q}^{\mathrm{I}}$. Since q_x^I depends only on x, and q_y^{I} depends only on y, the streamline can be found analytically [47]: Assuming $F_{x0} \neq F_{x1}$ and $F_{y0} \neq F_{y1}$, integration of each of the equations in (7) yields two separate expressions for the travel time along the streamline as a function of x and y, respectively,

$$t_x = \frac{1}{F_{x1}-F_{x0}} \ln\left(\frac{q_x^{\mathrm{I}}(x)}{q_x^{\mathrm{I}}(x_0)}\right), \tag{8}$$

$$t_y = \frac{1}{F_{y1}-F_{y0}} \ln\left(\frac{q_y^{\mathrm{I}}(y)}{q_y^{\mathrm{I}}(y_0)}\right). \tag{9}$$

By inserting $x = 0$ and $x = 1$ in (8) and $y = 0$ and $y = 1$ in (9), we determine the times t_{x0}, t_{x1}, t_{y0}, and t_{y1}, respectively, when the streamline crosses the corresponding straight lines. (Notice that these times may be negative or infinite). From these four travel times we can easily determine the exit time t_e when the streamline leaves the unit square. The exit point $\boldsymbol{x}_e$ is then found by inserting t_e in (8) and (9).

Applying Pollock's Method in Reservoir Simulation. For flow in porous media, the flow velocity $\boldsymbol{q}$ is obtained by solving (2) and (4) using e.g., a flux continuous scheme [1] to provide fluxes on each grid cell edge. In order to obtain particle velocities, these fluxes should be divided by the porosity. We assume for the moment that the grid cells can be any quadrilateral. The fluxes will be used to define a velocity field $\boldsymbol{q}^{\mathrm{I}}$ that approximates $\boldsymbol{q}$.

The flux is the integral of the normal component of the velocity field across an edge. We will require that $\boldsymbol{q}^{\mathrm{I}}$ reproduces the given fluxes,

$$F_E = \int_E \boldsymbol{q}^{\mathrm{I}} \cdot \boldsymbol{\nu}_E ds, \tag{10}$$

where the subscript E refers to the edge, F_E is the flux over the edge, and $\boldsymbol{\nu}_E$ is a unit normal to the edge. At the common edge between two adjacent grid cells, the absolute value of the flux is the same seen from both cells. By defining $\boldsymbol{\nu}_E$ suitably, we can assure that the sign of the flux is also the same. Thus only one flux is needed per cell edge.

We next assume that the normal component $(\boldsymbol{q}^{\mathrm{I}} \cdot \boldsymbol{\nu}_E)$ is constant along a given edge. Then (10) becomes

$$F_{E_i} = (\boldsymbol{q}^{\mathrm{I}}|_{E_i} \cdot \boldsymbol{\nu}_{E_i})\,|E_i|, \qquad i = 1, 2, 3, 4, \tag{11}$$

where E_i is one of the four edges, $\boldsymbol{q}^{\mathrm{I}}|_{E_i}$ denotes $\boldsymbol{q}^{\mathrm{I}}$ evaluated at a point on the edge, and $|E_i|$ is the length of the edge. Using this, combined with (10) for a unit square, gives

$$\begin{aligned} q_x^{\mathrm{I}}(0,y) &= F_{x0}, & \qquad q_y^{\mathrm{I}}(x,0) &= F_{y0}, \\ q_x^{\mathrm{I}}(1,y) &= F_{x1}, & \qquad q_y^{\mathrm{I}}(x,1) &= F_{y1}, \end{aligned}$$

where q_x^{I} and q_y^{I} are the x- and y-components of $\boldsymbol{q}^{\mathrm{I}}$, respectively. We are now in a position to introduce the velocity interpolation (6) and use Pollock's method to obtain the streamlines, as described above.

Finally, note that the interpolation step is not always necessary since (6) can be obtained directly when solving the pressure equation with a mixed finite-element method using the lowest-order Raviart-Thomas (RT_0) elements [9].

3.2. **Streamline and Velocity in Curvilinear Coordinates.** Streamline tracing on irregular grid cells involves introducing a curvilinear coordinate system [36] for each grid cell. In this section we describe the coordinate transformation and how to express the velocity in curvilinear coordinates. This transformation of coordinates and velocity will be fundamental for the development of the CVI method. Additionally, it will be used to derive the SFM and EFM methods.

Consider a quadrilateral grid cell in physical space $\mathcal{P}$ given by the four corner points $\boldsymbol{x}_i = [x_i, y_i]$, $i = 1, \ldots 4$. By using the bilinear isoparametric

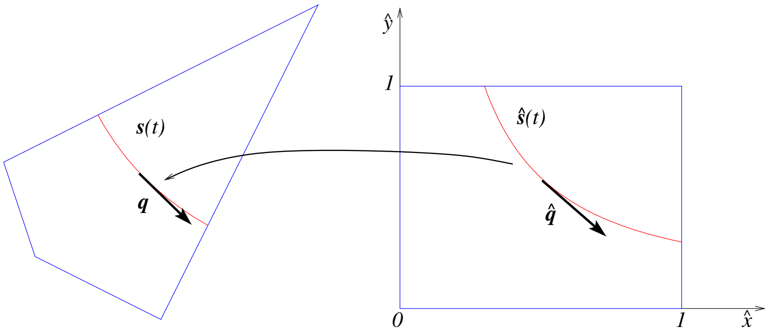

FIGURE 2. Transformation of a streamline and velocity from reference space $\mathcal{R}$ to physical space $\mathcal{P}$.

transformation [14, 15, 49]

$$\boldsymbol{x}(\hat{\boldsymbol{x}}) \equiv \sum_{i=1}^{4} \boldsymbol{x}_i \phi_i(\hat{x}, \hat{y}), \tag{12}$$

each grid cell is transformed into a unit square in the reference space $\mathcal{R}$. Here $\hat{\boldsymbol{x}} = [\hat{x}, \hat{y}]$ is a point in $\mathcal{R}$; $\boldsymbol{x}(\hat{\boldsymbol{x}}) = [x(\hat{x}, \hat{y}), y(\hat{x}, \hat{y})]$ is a point in physical space $\mathcal{P}$; and $\phi_i(\hat{x}, \hat{y})$, $i = 1, \ldots 4$, are the standard bilinear shape functions on the unit square.

Later we will compute normal vectors to the cell edges. In order to obtain a well-defined direction of these normals we require that $\boldsymbol{x}_i$, $i = 1, \ldots, 4$, are the logically bottom-left, bottom-right, top-left, and top-right corner of the quadrilateral, respectively.

A Velocity Transformation. If we can describe the velocity in bilinear coordinates, the streamline can be integrated in bilinear coordinates, and since each grid cell is a unit square in $\mathcal{R}$, Pollock's method is applicable. To obtain the streamline $\boldsymbol{s}(t)$ in $\mathcal{P}$, the bilinear transformation is applied to the streamline $\hat{\boldsymbol{s}}(t)$ in $\mathcal{R}$; see Figure 2. From this we can use the chain rule to deduce the velocity in $\mathcal{R}$,

$$\boldsymbol{q} \equiv \frac{d\boldsymbol{s}}{dt} = \frac{d\boldsymbol{x}(\hat{\boldsymbol{s}}(t))}{dt} = \frac{d\boldsymbol{x}}{d\hat{\boldsymbol{x}}}\frac{d\hat{\boldsymbol{s}}}{dt} = \boldsymbol{J}\hat{\boldsymbol{q}}. \tag{13}$$

Here $\hat{\boldsymbol{q}} = d\hat{\boldsymbol{s}}/dt$ is the velocity in $\mathcal{R}$, $\boldsymbol{q}$ is the velocity in $\mathcal{P}$, and $\boldsymbol{J} = d\boldsymbol{x}/d\hat{\boldsymbol{x}}$ is the Jacobian matrix of the transformation. Thus, the transformed velocity is given by

$$\hat{\boldsymbol{q}} = \boldsymbol{J}^{-1}\boldsymbol{q}. \tag{14}$$

Note that the Piola transformation [9, 30] of a vector field given by

$$\hat{\boldsymbol{q}}^{\mathrm{P}} = (\det \boldsymbol{J})\boldsymbol{J}^{-1}\boldsymbol{q} \tag{15}$$

is constructed so that fluxes are preserved in reference space. By comparing (14) and (15) we see that $\hat{\boldsymbol{q}}^{\mathrm{P}} = (\det \boldsymbol{J})\hat{\boldsymbol{q}}$.

3.3. **Extending Pollock's Method to Irregular Grids.** Tracing on irregular grids is done in reference space $\mathcal{R}$ using (14). The Jacobian matrix is given by

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \hat{x}} & \dfrac{\partial x}{\partial \hat{y}} \\ \dfrac{\partial y}{\partial \hat{x}} & \dfrac{\partial y}{\partial \hat{y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_x & \boldsymbol{u}_y \end{bmatrix}, \tag{16}$$

where $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ are the base vectors of the bilinear coordinates. Thus, the inverse of the Jacobian matrix can be expressed in terms of contravariant vectors, $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$, as

$$\boldsymbol{J}^{-1} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{n}_x^{\mathrm{T}} \\ \boldsymbol{n}_y^{\mathrm{T}} \end{bmatrix}, \tag{17}$$

where

$$\boldsymbol{n}_x = [\partial y/\partial \hat{y}, -\partial x/\partial \hat{y}]^{\mathrm{T}}, \qquad \text{and} \qquad \boldsymbol{n}_y = [-\partial y/\partial \hat{x}, \partial x/\partial \hat{x}]^{\mathrm{T}}. \tag{18}$$

The contravariant vectors, $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$, are normal vectors to edges of the quadrilateral in physical space $\mathcal{P}$. These vectors are generally not constant, but when evaluated at a particular edge of the quadrilateral, they are constant; and the length of these vectors is then equal to the length of the edge. It follows from (14) and (17) that,

$$\hat{\boldsymbol{q}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{q} \cdot \boldsymbol{n}_x \\ \boldsymbol{q} \cdot \boldsymbol{n}_y \end{bmatrix}. \tag{19}$$

Next, we approximate $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$ by $\boldsymbol{q}^{\mathrm{I}}$ and $\hat{\boldsymbol{q}}^{\mathrm{I}}$, respectively, based on the given fluxes in $\mathcal{P}$. Still using (11) for $\boldsymbol{q}^{\mathrm{I}}$, and recognizing the dot products in (19) as fluxes since $\boldsymbol{n}_E = \boldsymbol{\nu}_E |E|$, we can define $\hat{\boldsymbol{q}}^{\mathrm{I}}$ by the use of a linear flux interpolation,

$$\hat{\boldsymbol{q}}^{\mathrm{I}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}. \tag{20}$$

This expression is the basis for the standard flux-mapping (SFM) method [49] and the extended flux-mapping (EFM) method of Jimenez et al. [28]. In the SFM method [49], the bilinear Jacobian determinant is replaced by a constant value in order to make analytical integration possible. Hence,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{SFM}} = \frac{1}{\det \boldsymbol{J}_m} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}, \tag{21}$$

where $\boldsymbol{J}_m = \boldsymbol{J}(0.5, 0.5)$ is the Jacobian matrix evaluated at the midpoint of the reference element. In [43] it was shown that (20) is exact for uniform flow in 2D. Since the Jacobian determinant only scales the absolute value of the velocity in (21), SFM reproduces the shape of the streamlines exactly for uniform flow. However, approximating the Jacobian determinant by a constant introduces errors in computing time-of-flight, as noted by Jimenez et al. [28]. They demonstrated that by using a pseudo time-of-flight τ, the velocity in (20) can be integrated analytically by rewriting (20) as

$$\left\{ \begin{aligned} \frac{d\hat{x}}{F_{x0}(1-\hat{x}) + F_{x1}\hat{x}} &= \frac{dt}{\det \boldsymbol{J}} = d\tau, \\ \frac{d\hat{y}}{F_{y0}(1-\hat{y}) + F_{y1}\hat{y}} &= \frac{dt}{\det \boldsymbol{J}} = d\tau. \end{aligned} \right. \tag{22}$$

Then Pollock's method is used to find $\hat{x}(\tau)$ and $\hat{y}(\tau)$ and the exit pseudo time τ_e. To find the real exit time, t_e, we integrate the determinant of the Jacobian

$$t_e = \int_0^{t(\tau_e)} dt = \int_0^{\tau_e} \det \boldsymbol{J}(\hat{x}(\tau), \hat{y}(\tau)) d\tau. \tag{23}$$

Thus, the EFM method is characterized by the velocity field $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}}$ given in (20), where the Jacobian determinant is evaluated exactly.

3.4. **Extending the SFM and EFM Methods to 3D.** We consider an irregular grid consisting of hexahedral grid cells with fluxes computed on the faces of each cell. Each hexahedron in physical space $\mathcal{P}$ will be defined as a one-to-one trilinear map [14, 15, 49] of a unit cube in a reference space $\mathcal{R}$. The trilinear map is given by

$$\boldsymbol{x}(\hat{\boldsymbol{x}}) = \sum_{i=1}^{8} \boldsymbol{x}_i \phi_i(\hat{x}, \hat{y}, \hat{z}), \tag{24}$$

where $\boldsymbol{x}_i$, $i = 1, \ldots, 8$ are the coordinates of the eight corners defining the grid cell, and $\phi_i(\hat{x}, \hat{y}, \hat{z})$, $i = 1, \ldots, 8$ are the standard trilinear shape functions on the unit cube. Note that these hexahedrons generally have curved surfaces.

We remark that the mapping gives a natural definition of the faces of the hexahedron in physical space. We define general x-, y-, and z-surfaces in $\mathcal{P}$ by

$$\begin{aligned} s_x(\hat{x}) &= \left\{ \boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \le \hat{y} \le 1, 0 \le \hat{z} \le 1 \right\}, \\ s_y(\hat{y}) &= \left\{ \boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \le \hat{x} \le 1, 0 \le \hat{z} \le 1 \right\}, \\ s_z(\hat{z}) &= \left\{ \boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \le \hat{x} \le 1, 0 \le \hat{y} \le 1 \right\}. \end{aligned} \tag{25}$$

For the primary faces, $s_x(0)$, $s_x(1)$, etc., we will also use the notation S_{x0}, S_{x1}, respectively.

The velocity in physical space $\mathcal{P}$ is related to the velocity in reference space $\mathcal{R}$ by (13). The Jacobian matrix of the transformation is written,

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} & \frac{\partial x}{\partial \hat{z}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} & \frac{\partial y}{\partial \hat{z}} \\ \frac{\partial z}{\partial \hat{x}} & \frac{\partial z}{\partial \hat{y}} & \frac{\partial z}{\partial \hat{z}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_x & \boldsymbol{u}_y & \boldsymbol{u}_z \end{bmatrix}. \tag{26}$$

Here $\boldsymbol{u}_x$, $\boldsymbol{u}_y$, and $\boldsymbol{u}_z$ are the covariant base vectors of the trilinear coordinates. It follows that the inverse of $\boldsymbol{J}$ can be expressed in terms of contravariant vectors $\boldsymbol{n}_x$, $\boldsymbol{n}_y$, and $\boldsymbol{n}_z$ such that

$$\boldsymbol{J}^{-1} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{n}_x^{\mathrm{T}} \\ \boldsymbol{n}_y^{\mathrm{T}} \\ \boldsymbol{n}_z^{\mathrm{T}} \end{bmatrix}, \tag{27}$$

where

$$\boldsymbol{n}_x = \boldsymbol{u}_y \times \boldsymbol{u}_z, \qquad \boldsymbol{n}_y = \boldsymbol{u}_z \times \boldsymbol{u}_x, \qquad \boldsymbol{n}_z = \boldsymbol{u}_x \times \boldsymbol{u}_y. \tag{28}$$

Therefore,

$$\hat{\boldsymbol{q}} = \boldsymbol{J}^{-1}\boldsymbol{q} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{q} \cdot \boldsymbol{n}_x \\ \boldsymbol{q} \cdot \boldsymbol{n}_y \\ \boldsymbol{q} \cdot \boldsymbol{n}_z \end{bmatrix}. \tag{29}$$

As in the 2D case, the EFM method of Jimenez et al. [28] is based on (29) in combination with a linear flux interpolation,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \\ F_{z0}(1-\hat{z}) + F_{z1}\hat{z} \end{bmatrix}, \tag{30}$$

where F_{x0} is the given flux over the face S_{x0} in $\mathcal{P}$ corresponding to $\hat{x}$=0, etc. We will later use the normal vectors at the six primary faces defined as,

$$\boldsymbol{n}_{x0}(\hat{y},\hat{z}) \equiv \boldsymbol{n}_x(0,\hat{y},\hat{z}), \qquad \boldsymbol{n}_{x1}(\hat{y},\hat{z}) \equiv \boldsymbol{n}_x(1,\hat{y},\hat{z}), \tag{31}$$

$$\boldsymbol{n}_{y0}(\hat{x},\hat{z}) \equiv \boldsymbol{n}_y(\hat{x},0,\hat{z}), \qquad \boldsymbol{n}_{y1}(\hat{x},\hat{z}) \equiv \boldsymbol{n}_y(\hat{x},1,\hat{z}), \tag{32}$$

$$\boldsymbol{n}_{z0}(\hat{x},\hat{y}) \equiv \boldsymbol{n}_z(\hat{x},\hat{y},0), \qquad \boldsymbol{n}_{z1}(\hat{x},\hat{y}) \equiv \boldsymbol{n}_z(\hat{x},\hat{y},1), \tag{33}$$

where $(\hat{x},\hat{y},\hat{z}) \in [0,1] \times [0,1] \times [0,1]$.

In the SFM method by Prévost et al. [49], the Jacobian in (30) is evaluated at the midpoint of the unit cube.

3.5. **Reproduction of Uniform Flow.** By uniform flow we refer to flow given by a constant velocity field $\boldsymbol{q}$. Obviously, uniform flow leads to straight and parallel streamlines, and therefore the time-of-flight is equal at all points having the same distance from the inflow boundary.

In 2D, the linear flux interpolation used by EFM is exact for uniform flow, as shown in [43]. However, this is not the case in 3D: in [43] it was shown that the flux of a uniform flow field will vary quadratically. This might lead one to believe that replacing the linear interpolation in (30) with a quadratic, would solve the problem.

The difficulty with any flux interpolation can be seen from the following argument: Consider for instance the surface S_{x0}. A normal vector $\boldsymbol{n}_{x0}$ to this face at $\boldsymbol{x}(0,\hat{y},\hat{z})$ is given in (31). The absolute value of $\boldsymbol{n}_{x0}$ equals the surface Jacobian, which in the case of planar faces is constant only for parallelograms. For non-planar faces the direction of the normal vector is not constant either. By inserting $\hat{x}$=0 in (29) and (30) we see that both SFM and EFM use the following approximation

$$\boldsymbol{q} \cdot \boldsymbol{n}_{x0} = F_{x0}. \tag{34}$$

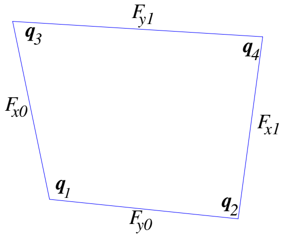

FIGURE 3. Reconstructing velocities from fluxes in 2D

For uniform flow, $\boldsymbol{q}$ is constant but $\boldsymbol{n}_{x0}$ will generally not be a constant, as noted above. The normal vector in (34) cannot change at a fixed point, so in effect we will trace the streamline using a velocity $\tilde{\boldsymbol{q}}$ such that $\tilde{\boldsymbol{q}} \cdot \boldsymbol{n}_{x0} = F_{x0}$, and $\tilde{\boldsymbol{q}}$ compensates for the fact that $\boldsymbol{n}_{x0}$ is not constant. Hence, the tracing velocity $\tilde{\boldsymbol{q}}$ will depend on the normal vector $\boldsymbol{n}_{x0}$, or in other words, depend on the geometry of the cell. Therefore uniform flow cannot be reproduced.

4. Corner Velocity Interpolation

In the previous section we described how the SFM and the EFM methods fail to reproduce uniform flow on e.g., grids with nonplanar faces. To remedy this problem, we will propose a different velocity interpolation scheme, which we will denote *corner velocity interpolation* (CVI). For simplicity, the method will first be introduced in 2D and then extended to 3D in Section 4.2.

4.1. **Interpolation in 2D.**

As for the SFM and EFM methods introduced in Section 3.3, we will formulate the CVI method using a cell-by-cell integration in the unit cube in reference space $\mathcal{R}$. However, the method may also be used to integrate streamlines directly in physical space; see [25] for more details. To formulate the CVI method, we start with the relation (19) for the velocity in $\mathcal{R}$, where the unknown quantity is the velocity $\boldsymbol{q}$ in $\mathcal{P}$. We approximate $\boldsymbol{q}$ by a bilinear interpolation of the velocities $\boldsymbol{q}_i$ at the corners $\boldsymbol{x}_i$, $i = 1, \dots, 4$,

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(\boldsymbol{x}(\hat{x}, \hat{y})) \equiv \sum_{i=1}^{4} \boldsymbol{q}_i \phi_i(\hat{x}, \hat{y}). \tag{35}$$

The corner velocities $\boldsymbol{q}_i$ will be reconstructed from the given fluxes, such that $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ is exact for uniform flow. This means that all $\boldsymbol{q}_i$ will be equal if $\boldsymbol{q}$ is constant.

Hence the CVI method is characterized by the following velocity interpolation in $\mathcal{R}$,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}} \equiv \boldsymbol{J}^{-1} \boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}. \tag{36}$$

Reconstruction of Corner Velocities. We consider the cell shown in Figure 3. The four fluxes F_i, will be given on the edges E_i for $i = x0, x1, y0, y1$, and the normal vectors are defined in (18). Note that $\boldsymbol{n}_x(\hat{x}, \hat{y}) = \boldsymbol{n}_x(\hat{x})$ and $\boldsymbol{n}_y(\hat{x}, \hat{y}) = \boldsymbol{n}_y(\hat{y})$. Hence, we can define $\boldsymbol{n}_{x0} = \boldsymbol{n}_x(0)$, $\boldsymbol{n}_{x1} = \boldsymbol{n}_x(1)$, $\boldsymbol{n}_{y0} = \boldsymbol{n}_y(0)$, and $\boldsymbol{n}_{y1} = \boldsymbol{n}_y(1)$, respectively.

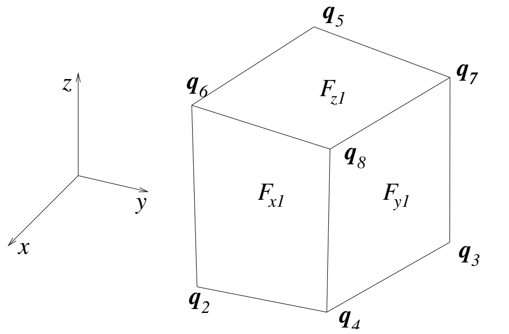

FIGURE 4. Reconstructing velocities from fluxes in 3D

The corner velocities $\boldsymbol{q}_i$, $i = 1, \ldots, 4$, will be solutions of 2×2 linear systems on the form

$$\left\{ \begin{array}{l} \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ex(i)} = F_{Ex(i)}, \\ \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ey(i)} = F_{Ey(i)}, \end{array} \right. \qquad i = 1, \ldots, 4. \tag{37}$$

Here $Ex(i)$ and $Ey(i)$, $i = 1, \ldots, 4$, refer to edges in the x- and y-direction, respectively, adjacent to corner $\boldsymbol{x}_i$. This means that for $\boldsymbol{q}_1$, we get

$$\left\{ \begin{array}{l} \boldsymbol{q}_1 \cdot \boldsymbol{n}_{x0} = F_{x0}, \\ \boldsymbol{q}_1 \cdot \boldsymbol{n}_{y0} = F_{y0}, \end{array} \right. \tag{38}$$

since E_{x0} and E_{y0} are adjacent to corner $\boldsymbol{x}_1$. The systems (37) are well-conditioned as long as the quadrilateral does not degenerate. If the fluxes have been computed exactly for a uniform flow field $\boldsymbol{q}$, then $\boldsymbol{q}_i = \boldsymbol{q}$.

Note that (37) implies that

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(E_i) \cdot \boldsymbol{n}_i = F_i, \qquad i = x0, x1, y0, y1, \tag{39}$$

where $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(E_i)$ denotes $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ evaluated at a point on edge E_i. Hence, since the length of $\boldsymbol{n}_i$ is equal to the length of E_i, $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ will reproduce the given edge fluxes.

4.2. **Extension to 3D.** We approximate $\boldsymbol{q}$ in (29) by a trilinear interpolation of the velocities $\boldsymbol{q}_i$ at the corners $\boldsymbol{x}_i$, $i = 1, \ldots, 8$,

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}} \equiv \sum_{i=1}^{8} \boldsymbol{q}_i \phi_i(\hat{x}, \hat{y}, \hat{z}). \tag{40}$$

Reconstructing Corner Velocities. Consider the cell in Figure 4. The six fluxes F_i will be given on the faces S_i for $i = x0, x1, y0, y1, z0, z1$, respectively. The corresponding normal vectors $\boldsymbol{n}_i$ are defined in (31).

The flux integral of a velocity field $\boldsymbol{q}$ over the face S_i can be transformed to a double integral on a face of the unit cube in $\mathcal{R}$ using the trilinear transformation:

$$F_i = \int_0^1 \int_0^1 \boldsymbol{q} \cdot \boldsymbol{n}_i d\alpha d\beta, \qquad i = x0, x1, y0, y1, z0, z1, \tag{41}$$

since the norm of $\boldsymbol{n}_i$ is equal to the surface Jacobian. If $\boldsymbol{q}$ is constant, we get,

$$F_i = \boldsymbol{q} \cdot \int_0^1 \int_0^1 \boldsymbol{n}_i(\alpha, \beta) d\alpha d\beta = \boldsymbol{q} \cdot \bar{\boldsymbol{n}}_i, \qquad i = x0, x1, y0, y1, z0, z1, \tag{42}$$

where $\bar{\boldsymbol{n}}_i$, defined by the above equation, is given by a simple analytic expression, see e.g., [1]. Thus, in order to reproduce a uniform flow field, the corner velocities $\boldsymbol{q}_i$, $i = 1, \ldots, 8$, will be solutions of 3×3 linear systems on the form

$$\begin{cases} \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sx(i)} = F_{Sx(i)}, \\ \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sy(i)} = F_{Sy(i)}, \\ \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sz(i)} = F_{Sz(i)}, \end{cases} \qquad i = 1, \ldots, 8. \tag{43}$$

Here $Sx(i)$, $Sy(i)$, and $Sz(i)$, $i = 1, \ldots, 8$, refer to faces in the x-, y-, and z-direction, respectively, adjacent to corner $\boldsymbol{x}_i$. That is, for $\boldsymbol{q}_8$ we get

$$\begin{cases} \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{x1} = F_{x1}, \\ \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{y1} = F_{y1}, \\ \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{z1} = F_{z1}. \end{cases} \tag{44}$$

Now, $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ and $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ are given by (40) and (36), respectively. As opposed to SFM and EFM, each component of the interpolated velocity field $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ is a function of all three variables $\hat{x}$, $\hat{y}$, and $\hat{z}$. Therefore, analytical integration of $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ is generally not possible.

Note that numerical integration of a velocity in $\mathcal{R}$ will not give the exact exit point from the cell, unless the last integration step ends on the cell boundary. Generally, interpolation is needed to determine the exit point [46].

4.3. **Comparison of SFM, EFM, and CVI.** Before summing up, let us compare the expressions of the EFM and CVI for the velocity field in 2D physical space. From Section 3.3 we know that

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}. \tag{45}$$

Furthermore, multiplication by $\boldsymbol{J}$ gives

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{EFM}} = \boldsymbol{J}\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} k_1\hat{x}\hat{y} + k_2\hat{x} + k_3\hat{y} + k_4 \\ k_5\hat{x}\hat{y} + k_6\hat{x} + k_7\hat{y} + k_8 \end{bmatrix}, \tag{46}$$

for certain coefficients k_i, $i = 1, \ldots, 8$, depending on the corners $\boldsymbol{x}_i$, $i = 1, \ldots, 4$, and the edge fluxes F_i, $i = x0, x1, y0, y1$. Note that for incompressible flow (for which $F_{x0} - F_{y1} + F_{y0} - F_{x1} = 0$), both k_1 and k_5 vanish. Some manipulations show that

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \sum_{i=1}^{4} (\det \boldsymbol{J}(\boldsymbol{x}_i)) \boldsymbol{q}_i \phi_i(\hat{x}, \hat{y}), \tag{47}$$

where $\boldsymbol{q}_i$, $i = 1, \ldots, 4$, are the corner velocities of the CVI method. Thus, the corner velocities for the CVI method and the EFM method are the same. It then follows that for parallelograms, where the determinant of the Jacobian

$\boldsymbol{J}$ is constant, the CVI and EFM are identical. Also, since the Jacobian determinant is bilinear, it can be written as

$$\det \boldsymbol{J} = \sum_{i=1}^{4} (\det \boldsymbol{J}(\boldsymbol{x}_i))\phi_i(\hat{x}, \hat{y}). \tag{48}$$

For a uniform flow field $\boldsymbol{q}$, we have, as noted after (38), that $\boldsymbol{q}_i = \boldsymbol{q}$. Therefore, it follows from (48) and (47) that $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{EFM}} = \boldsymbol{q}$. (Of course, by construction, the CVI method is also exact for uniform flow.) Equation (47) can only be extended to 3D for parallelepiped cells, since the corner velocities in 3D are determined from the average normals, $\bar{\boldsymbol{n}}_i$. The same is true for (48), since the Jacobian in 3D is not trilinear.

So far, we expect the main advantage of the CVI method to come with its extension to 3D. The CVI method is the only method that can reproduce uniform flow on irregular grids in 3D. In 2D, all the methods reproduce correct streamline paths for uniform flow, but only EFM and CVI reproduce the time-of-flight exactly. In the next section we demonstrate that CVI is the only method that converges for nonuniform grid refinements. We also show how the CVI method can be adapted to use half-edge fluxes. When the flow equations are solved by an MPFA method [1], half-edge fluxes are computed, and hence, more information about the velocity field is available. A disadvantage of CVI is that numerical integration must be used, whereas the EFM and SFM use analytical integration, which is faster.

5. Numerical Experiments

In this section we compare the three methods, SFM, EFM, and CVI with respect to accuracy in producing both streamline paths and time-of-flight. Moreover, we will discuss their relative computer efficiency.

Errors in a streamline tracing method may be classified in several categories. First, there may be errors in the computed velocity field or in the computed fluxes that are used as input data. This will be the case in Section 5.4, where we present results on two quarter-five spot configurations; on the unit cube and for a simplified real field model. The other numerical experiments however, consider only velocity fields with analytic representations to eliminate errors due to approximate velocities. For such velocity fields, the streamlines can be calculated analytically or numerically to any desired accuracy. These exact streamlines will then be used to measure the errors in the different tracing methods.

Disregarding round-off errors in e.g., reconstruction of the corner velocities from fluxes, the remaining errors may come from the following sources

- Errors in the transformation of the velocity from $\mathcal{P}$ to $\mathcal{R}$.
- Errors in the interpolation in $\mathcal{R}$. These are related to:
  - assumptions on the normal component of the velocity at cell edges, or
  - the interpolation method (linear/bilinear/trilinear).
- Errors in the integration due to:
  - the evaluation of the Jacobian determinant, or
  - numerical integration of the velocity, or

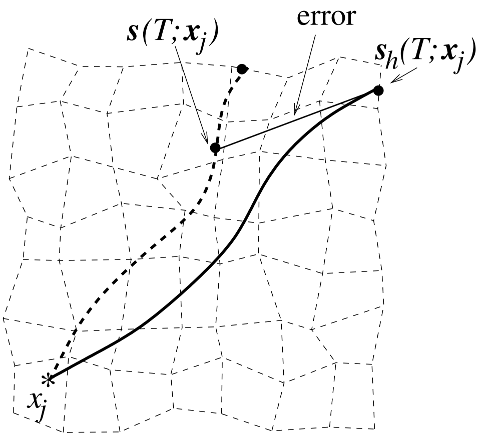

FIGURE 5. Error of the method

– interpolation to find the streamline exit points in the numerical integration.

These errors may combine or cancel each other at different parts of a given streamline. Due to the number of different error sources, and since the errors may cause the computed streamline to oscillate around, or diverge away from the true streamline, it is generally difficult to evaluate the accuracy and efficiency of a tracing method.

Error of the Tracing Method. Consider the exact streamline $\boldsymbol{s}(t; \boldsymbol{x}_j)$ and an approximate streamline $\boldsymbol{s}_h(t; \boldsymbol{x}_j)$ starting at the common point $\boldsymbol{x}_j$ at $t = 0$, see Figure 5. The length of the exact streamline at time t is given by

$$L(t) = \int_0^t \left| \frac{d\boldsymbol{s}(t; \boldsymbol{x}_j)}{dt} \right| dt. \tag{49}$$

The streamlines are traced until they reach the outflow boundary of the physical domain. Let T be the time-of-flight coordinate of the streamline that first arrives at a boundary. We choose to evaluate the error in the approximate streamline by (see [35, 46]):

$$\varepsilon_j(T) \equiv \frac{\|\boldsymbol{s}(T; \boldsymbol{x}_j) - \boldsymbol{s}_h(T; \boldsymbol{x}_j)\|}{L(T)}, \tag{50}$$

where $\| \cdot \|$ denotes Euclidean distance. This error measures the error per unit length in both streamline shape and time-of-flight.

The average error for a given method on a given grid for a number of starting points $\boldsymbol{x}_j$, $j = 1, \dots, N_s$, will be computed as

$$\bar{\varepsilon} = \frac{1}{N_s} \sum_{j=1}^{N_s} \varepsilon_j \tag{51}$$

Random Grids. In several of the test cases, irregular grids of hexahedral cells will be used. We start with a rectangular grid of $n_x \times n_y \times n_z$ uniformly partitioned grid cells. To obtain an irregular, randomly perturbed grid, each corner point in the uniform grid is perturbed by up to p percent relative to the grid cell size. Note that when these relative perturbations are retained for all refinements of the grid, these grids are referred to as rough grids. Refer to [34] for more details.

CVI Integration Method. To compute streamline paths, we have used the fourth-order, explicit Runge-Kutta solver in Matlab, ode45 [55]. This solver is based on the Dormand–Prince (4,5) pair [18], which uses six function evaluations per time step. The equation solved for each grid cell is

$$\frac{d\hat{\boldsymbol{s}}}{dt} = \hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}, \qquad \hat{\boldsymbol{s}}(0) = \hat{\boldsymbol{s}}_0. \tag{52}$$

Since this is a method with step size control, we can choose a relative tolerance δ_r and an absolute tolerance δ_a. In each time step, the solver estimates the local error $\boldsymbol{e} = [e_1, e_2, e_3]$ in the solution $\hat{\boldsymbol{s}} = [y_1, y_2, y_3]$. This error must be less than or equal to the acceptable error, which is a function of the specified relative and absolute tolerances

$$|e_i| \leq \max(\delta_r |y_i|, \delta_a), \qquad 1 \leq i \leq 3. \tag{53}$$

In the following we choose $\delta_a = 10^{-8}$ and use $10^{-8} \leq \delta_r \leq 10^{-3}$. If the time step is too large according to (53), the step is rejected, a smaller time step is computed, and a new set of function evaluations is needed. If the error is much smaller than the acceptable error, it is likely that the time step is unnecessarily small. The current step is therefore accepted and the next integration step is computed with a larger time step.

Generally, the last time step in a cell does not end on the cell boundary. Hence, interpolation is needed to determine the exit point [46]. The interpolation to find the exit point uses a fifth order Hermite polynomial [54, 55] and a Newton iteration.

Finally, it is important to choose a good initial step size to avoid an excessive number of function evaluations [6]. For our purposes, we will not go into this topic, but assume that a good initial step has been found. This is done by picking an initial step that is too large for the given tolerance δ_r, and letting the solver iterate until an acceptable step is found.

It is also possible to integrate streamlines directly in physical space based on CVI. However, using a higher-order Runge–Kutta method like ode45 directly in physical space will generally be computationally expensive, since these methods require the velocity to be evaluated in the interior of each cell. At such points, the velocity is only given in reference coordinates and a velocity evaluation thus requires an inverse bilinear/trilinear transformation that is quite computationally expensive. In [25], we therefore suggest an alternative method based on a simple Euler predictor-corrector scheme that only requires velocities at cell boundaries in combination with a grid refinement scheme.

Reference Streamlines. For comparing the tracing methods to an analytical solution, the exact streamline is computed using ode45 in Matlab with $\delta_r = 10^{-12}$ and $\delta_a = 10^{-12}$. The solution domain and the velocity is mapped to a reference element to simplify detection of domain boundaries.

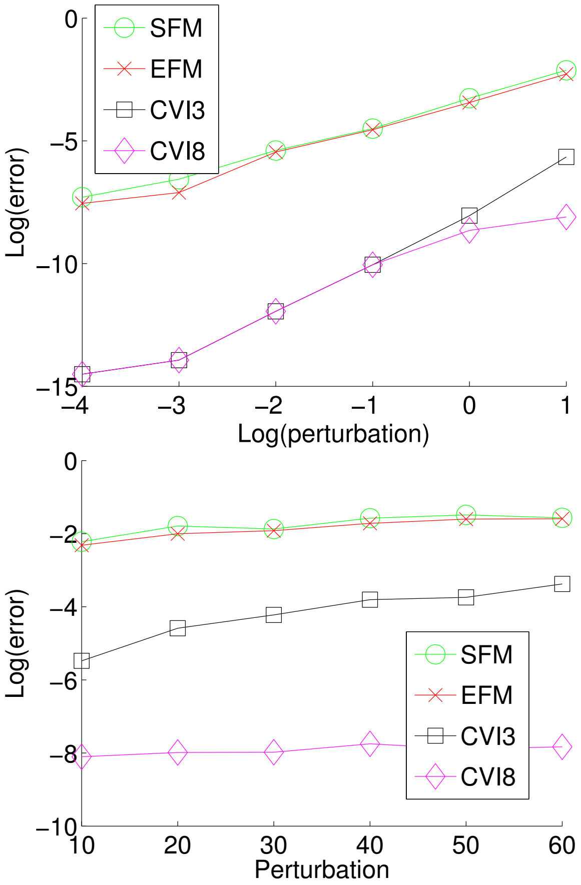

Figure 6. (Case 1, uniform flow). Logarithm of error relative to the degree of irregularity of cells. Top: $\log_{10}(p_k) = -4, -3, \dots, 1$. Bottom: $p_k = 10, 20, \dots, 60$.

5.1. **Uniform Flow on Random Grids.** We consider first four test cases for uniform flow on random grids using given constant velocity fields. Unless stated otherwise, the relative tolerance for the numerical integration in the CVI method is either $\delta_r = 10^{-3}$ or $\delta_r = 10^{-8}$, giving two methods denoted CVI3 and CVI8, respectively. The fluxes are computed analytically (to machine precision).

Case 1: Perturbed Grids. For the first test we consider a uniform flow field $\boldsymbol{q} = [1, 1, 1]^{\mathrm{T}}$, and a base $10 \times 10 \times 10$ Cartesian grid partition of the unit cube. The simulation will be performed on two series of perturbed grids, for which each vertex of the grid is moved a distance of up to p_k percent relative to the $h = 0.1$ grid spacing (i.e., on random grids with a p_k percent perturbation as defined in the previous subsection). For each grid and each tracing method, streamlines are traced from 100 random points in the front lower-left grid cell. The results of the test are shown in Figure 6. For this, and all subsequent plots, we have plotted the (base 10) logarithm of the average error given in (51).

Note that for a constant velocity field the Runge-Kutta solver will only need a single time step for each cell independent of the value specified for δ_r.

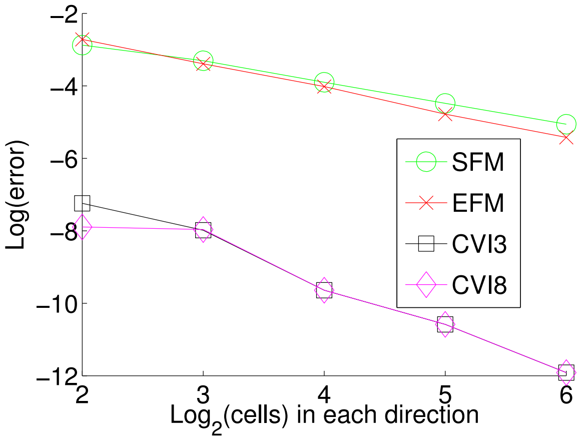

FIGURE 7. (Case 2, uniform flow). Logarithm of error relative to uniform grid refinement.

Thus, for velocity fields that are uniform or nearly uniform in a cell, we cannot assign any particular interpretation to the error tolerances used in the CVI method. However, since the velocity of the CVI method is integrated in reference space $\mathcal{R}$, this velocity will not be uniform in a cell even if the physical velocity field is uniform, unless the Jacobian matrix of the transformation is constant (i.e., the cells tend to parallelepipeds).

Figure 6 shows that for small grid perturbations the errors of the CVI methods may be below their prescribed error tolerances, indicating that the velocity field in $\mathcal{R}$ is also uniform. As we reach a 1% perturbation, we begin to see the effect of the chosen relative tolerances for the CVI method. The results indicate that the error for the CVI method can be made arbitrarily small by decreasing δ_r and δ_a.

Case 2: Uniform Refinement. For the second test we consider the error with respect to uniform grid refinement. We still consider a flow field $\boldsymbol{q} = [1,1,1]^{\mathrm{T}}$. Grid $\mathcal{G}_0$ consists of one grid cell and is a 50% random perturbation of the unit cube. Grid $\mathcal{G}_k$ for $k = 1, 2, \ldots, 6$ will be a trilinear map (see Section 3.4) of a uniform $2^k \times 2^k \times 2^k$ partition of the unit cube in $\mathcal{R}$ to the cell represented by $\mathcal{G}_0$, see Figure 8 (left). The grids $\mathcal{G}_k$ will thus be refinements of $\mathcal{G}_0$ and asymptotically tend to a parallelepiped grid.

Streamlines are traced from 100 random points in the front lower-left grid cell of the finest grid ($\mathcal{G}_6$). The average error for grid $\mathcal{G}_k$, $k = 2, \ldots, 6$, is shown in Figure 7.

We observe that the CVI method is both more accurate and converges faster than EFM and SFM. For an $8 \times 8 \times 8$ refinement ($\mathcal{G}_3$), EFM and SFM have an error of approximately 10^{-3}. The same accuracy is achieved for the initial grid $\mathcal{G}_0$ (not shown in the plot) for the CVI3 method. For the given velocity field, the streamlines for EFM and SFM on grid $\mathcal{G}_3$, are traced through 21 cells. Consequently, SFM and EFM have to trace approximately 21 cells to achieve the same accuracy that CVI3 obtains on a single cell.

Moreover, we observe that the errors for the CVI methods are almost independent of the tolerance for $\mathcal{G}_k$, $k \geq 3$. The reason is that as the

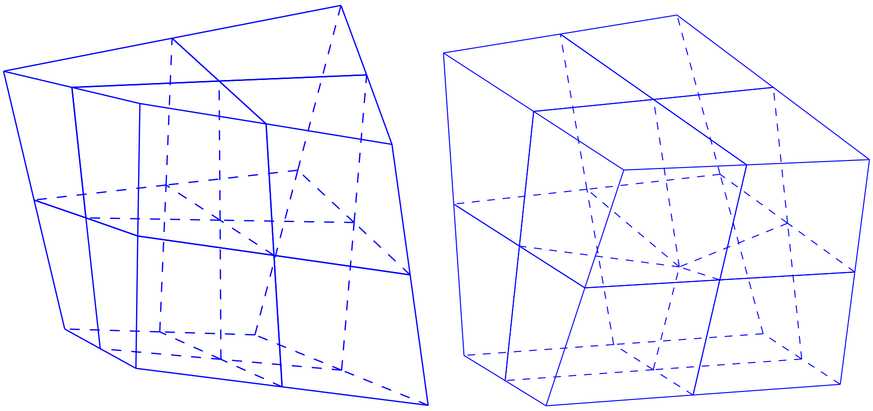

FIGURE 8. Grids for the first refinement level ($\mathcal{G}_1$) in Case 2 (left) and Case 3 (right). Note that the base grid $\mathcal{G}_0$ is different for the two figures, and also only internal grid points are perturbed for Case 3, i.e., the central corner in the figure.

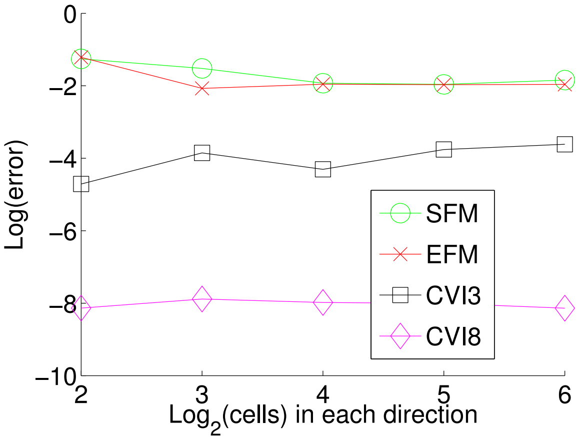

FIGURE 9. (Case 3, uniform flow). Logarithm of error relative to nonuniform grid refinement.

grid is refined and the grid cells tend to parallelepipeds, the velocity field in reference space also becomes approximately uniform, and the numerical integration will need only one time step for any δ_r.

Case 3: Nonuniform Refinement. The setup will be the same as for Case 2, except that the refinement now is random. At each refinement level, the grids will be a 50% perturbation of a uniformly refined grid, see Figure 8 (right). The average error for grid $\mathcal{G}_k$, $k = 2, \ldots, 6$ is shown in Figure 9. (The choice of a nonuniform refinement may seem strange, but is in fact what one will use for real field cases when trying to approach the resolution of an underlying geomodel, which will typically consist of highly irregular grid cells modeling complex geological structures.)

For this case, the velocity in reference space will not become uniform at the same rate as for the uniform refinement. Due to the nonlinearity of the velocity field in $\mathcal{R}$, the errors of the CVI method cannot be reduced further

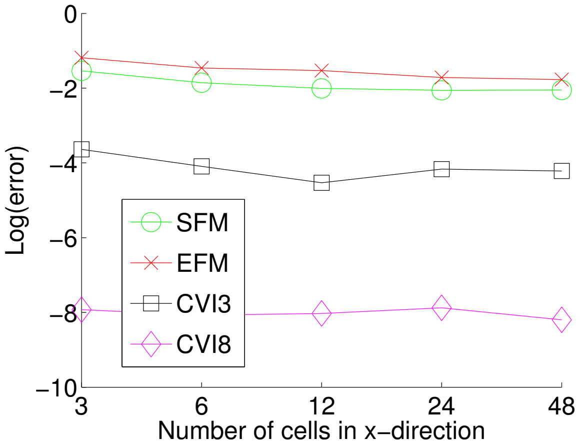

FIGURE 10. (Case 4, uniform flow). Logarithm of error relative to length of streamline.

than the prescribed tolerances used in the numerical integration; i.e., 10^{-3} for CVI3 and 10^{-8} for CVI8, respectively. The errors are therefore almost independent of the grid spacing. To increase the accuracy of CVI, one must select a lower tolerance δ_r, which implies an increased number of integration steps. The situation is quite different for the SFM and EFM methods: Here the horizontal curves indicate that these methods do not converge as the grid is refined!

Case 4. In this test we consider error accumulation due to the length of the streamline and the number of cells traversed. For this case we use the flow field $\boldsymbol{q} = [1, 0, 0]^{\mathrm{T}}$. Grid $\mathcal{G}_k$ for $k = 0, 1, \ldots, 4$ will be a 50% perturbation of a $3 \cdot 2^k \times 3 \times 3$ partition of the parallelepiped $[0, 2^k] \times [0, 1] \times [0, 1]$. Twenty streamlines are traced, each with a different random starting point in the central-left cell of each grid. Then the number of cells traversed for a given streamline for a given grid $\mathcal{G}_k$ will be approximately $3 \cdot 2^k$.

Figure 10 shows the error per unit length for each streamline. Since this error is approximately constant for each method, we can in other words expect a uniform increase in the *absolute error* $\|\boldsymbol{s}(T; \boldsymbol{x}_j) - \boldsymbol{s}_h(T; \boldsymbol{x}_j)\|$ with increasing streamline length for all methods. Moreover, since the error constant is larger for EFM and SFM than for the CVI method, the accumulation of error with streamline length will be more pronounced for SFM and EFM.

We also see from (21) that the only difference between EFM and SFM is the scaling of the velocity vector. Thus the streamline shape will be the same for both methods, and the differences in the figure are only due to different time-of-flights. EFM uses a more accurate approximation of the Jacobian determinant than SFM and should therefore in principle be more accurate. Here, however, SFM is more accurate than EFM, which is explained by error cancellation in which the error in approximating the Jacobian determinant cancels the error in the interpolation.

5.2. **Nonuniform Flow on Random Grids.** Streamline simulators uses a sequential splitting method to solve the flow equations as explained in Section 2. For an incompressible flow in an isotropic and homogeneous

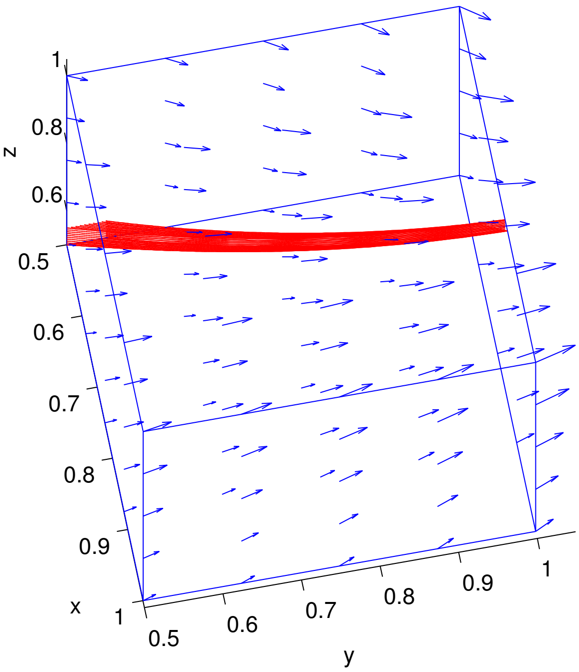

FIGURE 11. Field plot for the velocity in (55). All the analytical streamlines traced for this case lie within the streamtube indicated in red.

medium with no sources, (2) reduces to the well-known Laplace equation. For this case it is easy to verify that

$$p(x, y, z) = \sin(\sqrt{2}x)\sinh(y)\sinh(z) \tag{54}$$

is a solution to (2). The corresponding velocity field is given by (4) as

$$\boldsymbol{q} = -\begin{bmatrix} \sqrt{2}\cos(\sqrt{2}x)\sinh(y)\sinh(z) \\ \sin(\sqrt{2}x)\cosh(y)\sinh(z) \\ \sin(\sqrt{2}x)\sinh(y)\cosh(z) \end{bmatrix}. \tag{55}$$

We will now consider the same type of tests as for uniform flow, using the velocity field in (55). The setup for the test cases will be similar to the cases for uniform flow in the previous section, except for the following:

- The analytical velocity field is now nonlinear, and numerical integration must be used to obtain fluxes. We have used a Lobatto quadrature [21] with a tolerance of 10^{-6}.
- The domain is shifted slightly to avoid the singularity of the velocity at the origin, see Figure 11.
- Twenty streamlines are traced from the surface (see Figure 11)

$$S = \{(x, y, z) \; : \; x = 0.5, \; 0.5 \le y, z \le 0.55\}.$$

Case 1: Perturbed Grids. First we revisit the perturbed grids from Case 1 in Section 5.1, i.e., consider a p_k percent perturbation of the cells in a $10 \times 10 \times 10$ partition of the cube shown in Figure 11, but now with the velocity field (55). The results are shown in Figure 12. The curves for CVI8 are

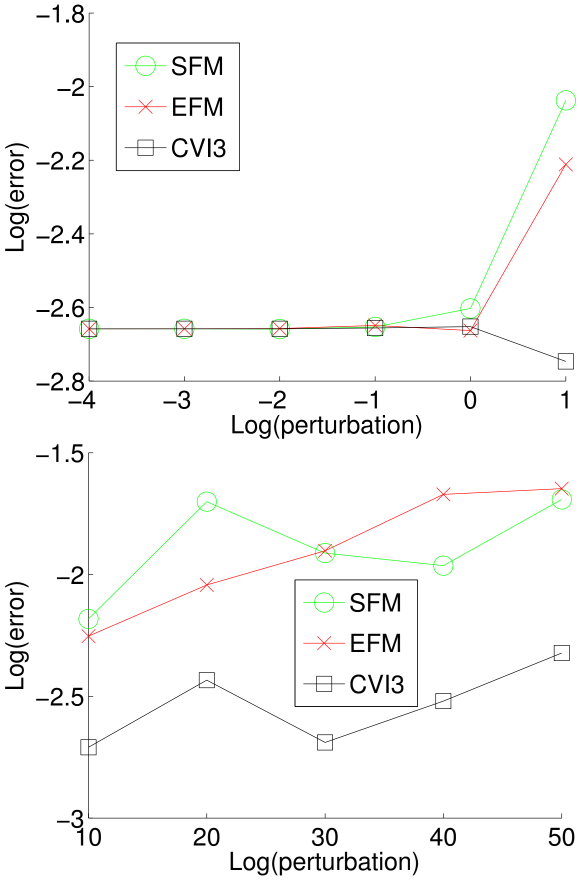

FIGURE 12. (Case 1, nonuniform flow). Logarithm of the error relative to the degree of irregularity of cells Top: $\log_{10}(p_k) = -4, -3, \ldots, 1$. Bottom: $p_k = 10, 20, \ldots, 50$.

identical to those of CVI3 and are therefore not shown in the plots. For small perturbations, the difference in the methods are overshadowed by the error in the interpolation since all methods use the same interpolation for small perturbations. As the perturbations increase, we see the differences in the methods. However, due to possible cancellation of errors (e.g., as discussed for Case 4 in Section 5.1), it is difficult to interpret these differences in favor of one method or the other, even though CVI generally has a slightly lower error than SFM and EFM.

Case 2: Uniform Refinement. Next, we consider the error relative to a uniform refinement of a single skewed cell, as in Case 2 of Section 5.1. The results are shown in Figure 13, where the CVI8 curve (not shown) is identical to the CVI3 curve. The figure shows that the convergence rates are equal for all three methods, indicating that the perturbations of the cells go faster to zero than the velocity approaches uniform flow in a given cell (otherwise, the CVI method would converge faster than SFM/EFM).

Case 3: Nonuniform Refinement. Corresponding results for nonuniform refinement are shown in Figure 14. This test case shows again the benefit of

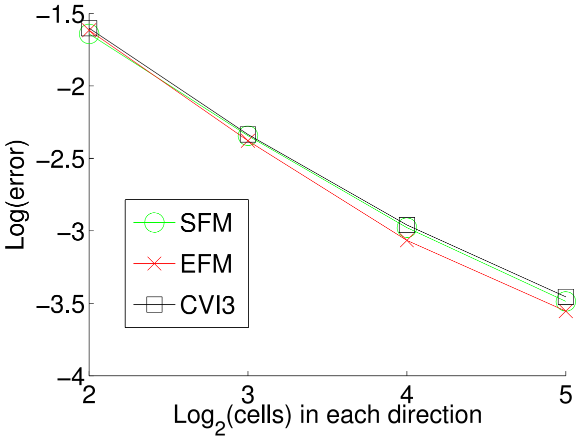

FIGURE 13. (Case 2, nonuniform flow). Logarithm of error relative to uniform grid refinement.

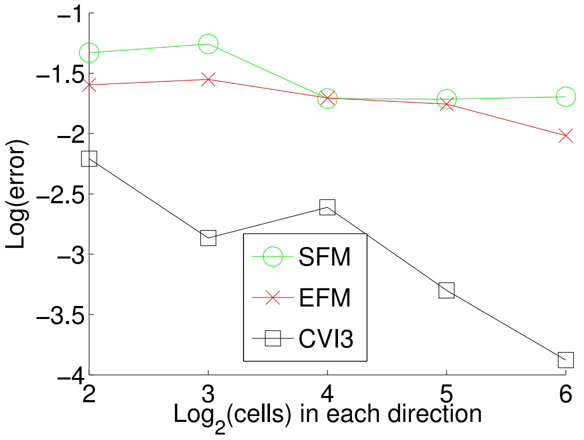

FIGURE 14. (Case 3, nonuniform flow). Logarithm of error relative to nonuniform grid refinement.

the CVI method when the flow becomes approximately uniform in each cell and the grid cells are kept irregular: As for Case 3 in Section 5.1 for uniform flow, SFM and EFM do not seem to converge as the grid is refined.

Note, however, that the curves depend on a particular realization of the random grids. Ideally, we should have averaged over a set of such realizations. This may explain the irregular behavior of the CVI curve. Also, since the fluxes are computed numerically with a tolerance of 10^{-6}, we cannot expect the same level of accuracy as the grid is refined for the CVI methods compared to Case 3 for uniform flow, where the fluxes were computed analytically.

5.3. **Analytic Flow on Truncated Pyramidal Grids.** We now study the effect of using a special kind of grid, denoted a *truncated pyramidal grid*, see also [43]. The construction of such a grid is illustrated by Figure 15, where the entire grid conforms to a cubic domain. As seen here, the inner cells are truncated pyramids, and 'infill' cells are used to account for the boundary,

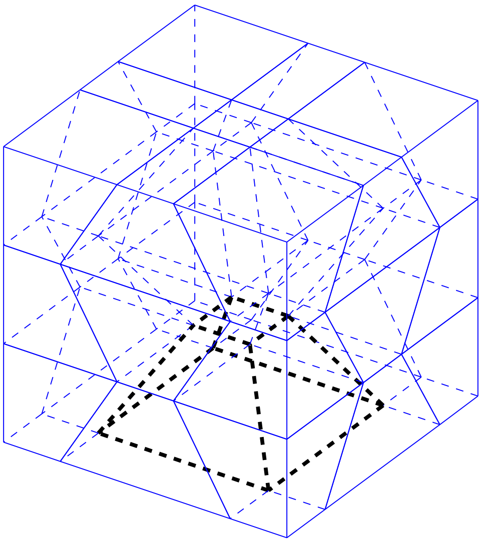

FIGURE 15. Sketch of a truncated pyramidal grid

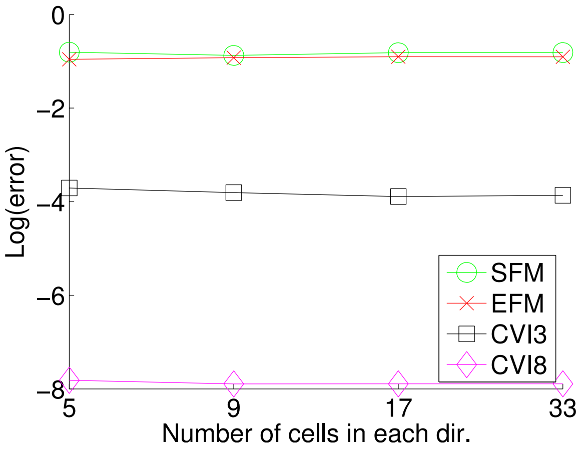

FIGURE 16. (Case 1, truncated pyramids). Errors versus grid refinement.

and the cells will be turned upside down systematically from layer to layer. The surface with the smallest area of a truncated pyramid will either be the roof surface or floor surface of the cells, and the ratio between the smallest and the largest surface area is kept constant and equal to 1/9 as the grids are refined.

As pointed out in the description of the CVI method (Section 4), this method is constructed to reproduce uniform flow exactly regardless of the grid geometry. SFM and EFM, on the other hand, do not reproduce uniform flow and will therefore, as observed from the results below, produce a systematic error for skewed cells like the truncated pyramids.

Case 1: Uniform Flow. Figure 16 shows the performance of the three tracing methods on a sequence of refinements of truncated pyramidal grids for the uniform flow field $\boldsymbol{q} = [0, 0, 1]$ on the cubic domain $\{1 \leq x, y, z \leq 2\}$.

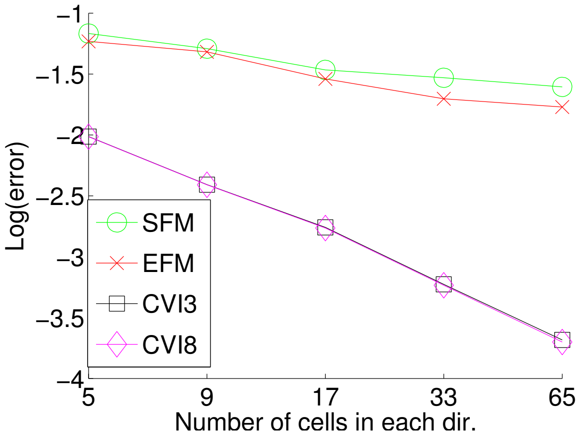

Figure 17. (Case 2, truncated pyramids). Errors of streamlines versus grid refinement.

For each refinement, one hundred streamlines are traced from randomly distributed points on the bottom surface ($z = 1$) of the domain until they exit at the top of the domain ($z = 2$). We see that the errors of the CVI method stay at the level of their prescribed tolerances, indicating that the velocity in the reference space is not becoming uniform as the grid is refined (compare with Figure 7). We also notice that the streamlines of SFM and EFM do not converge as the grid is refined.

Case 2: Combined Uniform and Nonuniform Flow. We next replace the uniform flow field by a combination of nonuniform and uniform flow, obtained by taking the negative gradient of the pressure field

$$p(x, y, z) = \cosh(x)\cos(y) + cz,$$

which is an analytical solution to the Laplace equation that here represents a simplification of the general pressure equation (2). Note that this pressure field gives rise to a uniform flow in the z-direction since the z-component of the gradient of the pressure field is a constant. This situation can occur if gravity is the driving force for the flow in the reservoir and the permeability of the medium is almost homogeneous. Then the flow locally is almost uniform in the z-direction of the reservoir.

Figure 17 depicts how the methods behave for refinements of the truncated pyramidal grids introduced above, where the constant $c = 5$ is used and twenty streamlines are traced for each refinement level. The CVI method converges with the expected rate, whereas the convergence for SFM and EFM is both slower and decays significantly for the highest refinements, suggesting that these two methods may fail to converge asymptotically.

Case 3: Nonuniform Flow. Finally, we study the effect of a gradual change from uniform flow in the z-direction to nonuniform flow in all directions. This is done by allowing the σ-parameter to increase for the solution

$$p(x, y, z) = \cosh(x)\cos(y) + cz + \sigma\cosh(x)\cos(z). \tag{56}$$

The result for an $11 \times 11 \times 11$ grid is presented in Figure 18, where we have plotted the $\log_{10}\sigma$ along the x-axes. As expected, the errors decrease

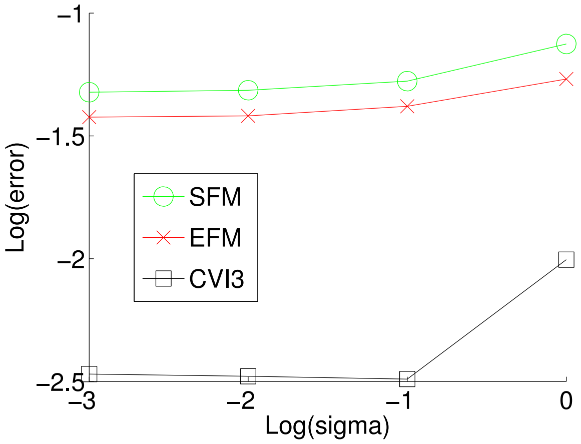

FIGURE 18. (Case 3, truncated pyramids). Effect of flow becoming gradually more nonuniform by increasing σ in Equation (56) when the grid resolution is fixed.

as the flow approaches uniform in the z-direction (i.e., $\sigma \to 0$). Repeating the refinement process from Case 2, we observed that the CVI method still converges for a fixed σ and increasing refinements, but the error constant increases for increasing values of σ.

Summing up, when either truncated pyramids or general rough grids are used in 3D, the difference between our new streamline method and SFM and EFM is striking. The CVI method is superior to the other methods in terms of accuracy; this is manifested as either loss of convergence or diminished convergence rates for the SFM and EFM methods.

5.4. **Quarter-Five Spot.** In this subsection we use a control-volume finite-difference method [1] to solve the single-phase flow equations,

$$\begin{aligned} \operatorname{div} \boldsymbol{q} = -\operatorname{div}(\boldsymbol{K} \operatorname{grad} p) &= f \text{ in } \Omega \\ \boldsymbol{q} \cdot \boldsymbol{n} &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{57}$$

and provide discrete fluxes on cell interfaces. Here, $\boldsymbol{q} = -\boldsymbol{K} \operatorname{grad} p$ is the Darcy velocity, p is the pressure, $\boldsymbol{K}$ is the permeability tensor, f is a source term, Ω is the solution domain, $\partial\Omega$ is its boundary, and $\boldsymbol{n}$ is the unit outward normal to $\partial\Omega$. The domain will be discretised with different kind of $n_x \times n_y \times n_z$ grids.

A 3D extension of the classical 2D quarter-five spot test case is generated by placing an injector and a producer, respectively, in the lower and upper corner cells $(1, 1, 1)$ and (n_x, n_y, n_z). Streamlines will then be traced from the injector to the producer. As an error measure, we will in this subsection only compare the time-of-flight in the producing wells. To this end, we use a relative error measure by dividing the absolute difference in time-of-flights to the time-of-flight of the reference streamline. These relative errors are then averaged over all the traced streamlines.

Case 1: Random Grids. Let Ω be the unit cube and assume that $\boldsymbol{K} \equiv 1$ over the whole domain. On this simplified medium, we compare the streamline

Table 1. Average errors and standard deviations for the different tracing methods. Random grid perturbations of 16%.

Method	Average error	St.dev
SFM	0.022	0.015
EFM	0.021	0.015
CVI3	0.017	0.012

methods on a coarse $10 \times 10 \times 10$ random grid. The inner corners of the coarse grid are perturbed such that the grid is non-orthogonal throughout the simulation domain. Wells are implemented by using non-homogeneous Neumann conditions at the faces of the corner cells $(1, 1, 1)$ and $(10, 10, 10)$. We trace streamlines starting from 225 uniformly spaced entry points along the inflow boundary given by the boundary faces of cell $(1, 1, 1)$. The streamlines are traced until they exit at the outflow boundary, i.e., at some point of the boundary faces of cell $(10, 10, 10)$.

Time-of-flights at the producer are compared to the reference case simulated by the SFM method on a $50 \times 50 \times 50$ uniform Cartesian grid, where the entry points agree with the entry points of the coarse grid. Table 1 reports the average errors and corresponding standard deviations for the three tracing methods. As seen from the table, the CVI method performs slightly better than the other two methods; the average error is roughly 20% smaller for the CVI method compared to e.g., SFM.

Case 2: A Simplified Field Case. Next, we study the streamline generation on a simplified version of a real field model. We use a 3D grid that is an extension of a 2D grid where the height of the top and bottom of the medium varies throughout. The grid has moderate grid aspect ratios that resemble typical features of field cases; here this ratio is approximately 1/20 for height versus length of a typical grid block. A reference solution will be generated on a fine $45 \times 45 \times 45$ grid, which is a uniform refinement of a coarse $9 \times 9 \times 9$ grid on which the tracing methods will be compared. The two grids are compatible in the sense that each cell interface of the coarse grid is exactly matched by a set of interfaces in the fine grid. To compute a reference solution we solve the flow equations on the fine grid with no-flow boundary conditions and a nonzero source f to simulate wells. The approximate solution on the coarse grid are found by simple averaging of fine-grid fluxes.

We will apply two different permeability fields; first a homogeneous permeability field, and then a layered lognormal permeability field that has similarity to real field cases. For both cases, three hundred streamlines are traced from the injector to the producer. The reference solution will be traced with SFM on the $45 \times 45 \times 45$ grid.

For the homogeneous case, $\boldsymbol{K} \equiv 1$, averaged errors and associated standard deviations for the three methods are presented in Table 2. We see that there is an apparent improvement for CVI compared to SFM and EFM, and both errors and standard deviations are significantly smaller.

TABLE 2. Average errors and associated standard deviations for the three tracing methods for the homogeneous case.

Method	Average error	St.dev
SFM	0.084	0.076
EFM	0.083	0.077
CVI3	0.077	0.068

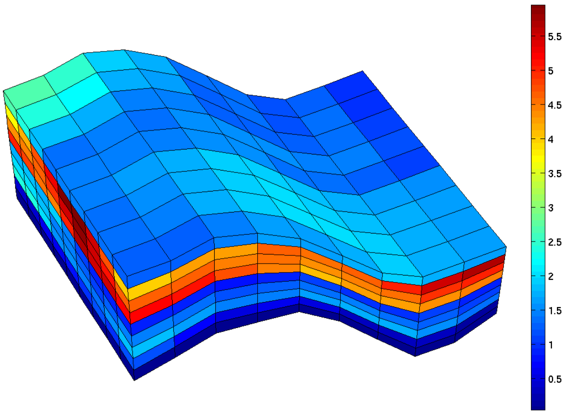

FIGURE 19. Logarithm of the horizontal permeability for the simplified 3D field model.

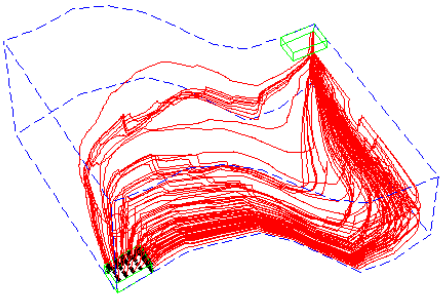

FIGURE 20. Seventy-five streamlines traced using the fine-grid velocity for the layered permeability field shown in Figure 19. Injector and producer cells for the coarse grid are shown in green. Black dots indicate starting points for the streamlines.

For the heterogeneous case, the permeability field is defined on the coarse $9 \times 9 \times 9$ grid. On the fine grid, the permeability is therefore constant on patches of $5 \times 5 \times 5$ cells. We apply an isotropic, layered log-normally distributed permeability field, see Figure 19. Figure 20 depicts seventy-five of the streamlines traced on the fine grid, and Figure 21 shows the

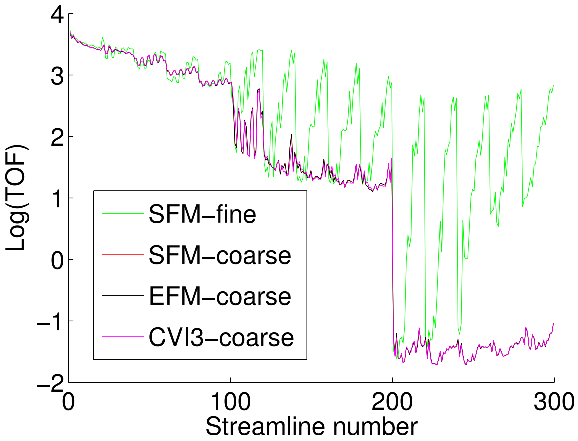

FIGURE 21. Time-of-flight distribution for SFM on the fine grid, and for SFM, EFM, and CVI3 on the coarse grid. The EFM and CVI3 results on the fine grid are not shown since they could not be distinguished from the SFM curve on the fine grid.

base 10 logarithm of the time-of-flight for the reference solution and for the approximate solutions.

Note that in Figure 21 the red SFM curve and the black EFM curve are more or less covered by the magenta CVI3 curve. Due to the large variation in permeability on the coarse $9 \times 9 \times 9$ grid, we observe a huge variations in time-of-flight for different streamlines. Unfortunately, there are equally large differences if the same streamline is traced using coarse and fine-grid fluxes, respectively, which indicates nonlinearity in the velocity field and a high information loss in the flux averaging. This shows the futility of comparing different streamline methods on 'realistic' models, that is for skew grids and permeability fields with complex heterogeneity structures. In this paper, we have therefore mainly focused on simplified models, on which one has control of the reference solution and errors in the velocity fields on the coarser grids. More details can be found in [44], where a discontinuous Galerkin method was used for computing time-of-flight.

5.5. **Runtime Comparisons.** Finally, we investigate the runtimes of the different methods. For the CVI method, there will be a relation between the number of time steps needed per cell and the prescribed tolerance δ_r. Tables 3 and 4 report the average number of time steps per cell for different values of δ_r and different perturbations of the grids for Case 1 from Sections 5.1 and 5.2.

Timing Results. Let us now compare the computational costs required to determine the exit point given a starting point within the cell. If we normalize the results such that SFM uses one time unit on average, EFM uses 1.5 time units, and CVI uses eight time units (assuming one time step per cell).

The runtime of CVI is essentially made up of the time used for:

- function evaluations,

Table 3. Relation between number of time steps per cell and δ_r for a 60% perturbation for Case 1 for uniform and nonuniform flow.

Tolerance δ_r	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Uniform	1.3	1.7	2.3	3.0	4.2	5.2
Nonuniform	1.4	1.9	2.5	3.2	4.4	5.4

Table 4. Relation between cell perturbation and number of time steps per cell for a given δ_r for Case 1 for uniform and nonuniform flow.

Perturbation in %	0	10	20	30	40	50	60
Uniform ($\delta_r = 10^{-3}$)	1.0	1.0	1.0	1.1	1.1	1.2	1.3
Uniform ($\delta_r = 10^{-8}$)	1.0	2.3	3.2	3.6	4.2	4.6	5.2
Nonuniform ($\delta_r = 10^{-3}$)	1.0	1.0	1.0	1.0	1.1	1.2	1.4
Nonuniform ($\delta_r = 10^{-8}$)	1.1	2.2	3.0	3.4	3.8	4.4	5.4

- interpolation to find the exit point, and
- the reconstruction of the corner velocities.

By a function evaluation we mean evaluation of the velocity at a given point in reference space. This is done, as shown in Section 4, by a trilinear interpolation to find the velocity in physical space. Then the velocity in reference space is found by a multiplication by the inverse of the Jacobian matrix. The evaluation of the inverse is quite time-consuming, so a function evaluation takes approximately one time unit. One integration step generally requires six function evaluations, but for the first integration step in a cell, we need one extra function evaluation to get started. The interpolation to find the exit point uses a fifth-order polynomial [55] and a Newton iteration. This part consumes approximately one half time unit. Finally, reconstruction of corner velocities from discrete fluxes also takes one half time unit. Altogether, eight time units.

Although the CVI method is slower, the accuracy is higher when the flow becomes uniform. Our test cases for uniform flow indicate that to reach the same level of accuracy, SFM and EFM must use a refined grid, for which the computational cost is higher, in particular if refining the grid requires a new global pressure solution since the cost of streamline tracing typically is minuscule compared with the cost of solving the pressure equation. Moreover, in [25] we showed that by integrating directly in physical space rather than in reference space, the CVI method can be made as efficient as EFM for cases with uniform and almost uniform flow in 2D.

5.6. **Using CVI with Half-Edge Fluxes.** The CVI method interpolates velocities in a manner that makes it appealing to use a *finer resolution* of the velocities. Moreover, if fluxes/velocities are evaluated for the half-edge

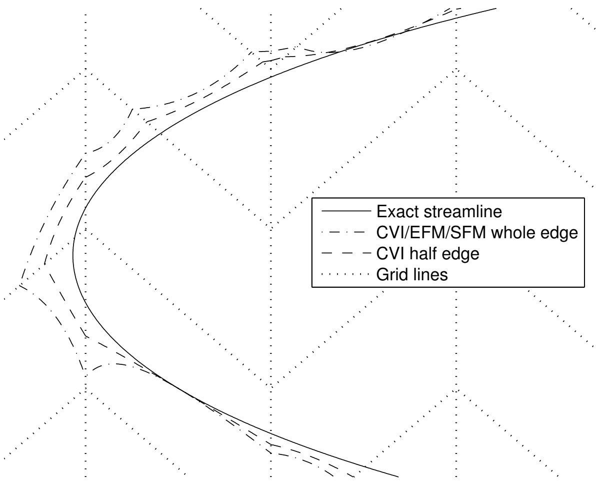

FIGURE 22. Using CVI with half-edge fluxes. Note that the aspect ratio of the plot is 1:2.75, i.e., the y-direction is compressed. A horizontal line really makes an angle of 71° with the bottom or top edge of the parallelograms.

rather than for the full edge, this subresolution in the fluxes may possibly be taken advantage of.

We have tested the CVI method with half-edge fluxes (denoted CVIH) in 2D for the velocity field,

$$\boldsymbol{q} = \begin{bmatrix} \sinh x \cos y \\ -\cosh x \sin y \end{bmatrix}, \tag{58}$$

which is an analytical solution to the Laplace equation. When using CVI with half-edge fluxes, the interpolated velocity field is still given by (35), but the corner velocities will be different. Equation (37) is replaced by

$$\begin{cases} \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ex(i)} = 2F_{Ex(i)}, \\ \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ey(i)} = 2F_{Ey(i)}, \end{cases} \qquad i = 1, \dots, 4, \tag{59}$$

where $F_{Ex(i)}$ and $F_{Ey(i)}$ are now *half*-edge fluxes adjacent to corner i. Equation (59) implies that the normal component of $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ varies *linearly* along each edge.

Case 1: Half-Edge Fluxes on Parallelogram Grids. For parallelogram grids, CVI, SFM, and EFM are identical, as noted in Section 4.3. When the skewness of the parallelograms gets large, these methods will produce streamlines with apparent cusps. The CVIH method will reduce these artifacts, see Figure 22.

Case 2: Comparing BDM and CVIH. A mixed finite-element method [9] can be employed to solve the pressure equation for both pressure and velocity, in which case the velocity must belong to the space $H(\mathrm{div};\Omega)$. A divergence-free higher-order velocity field can then be used to approximate $H(\mathrm{div};\Omega)$.

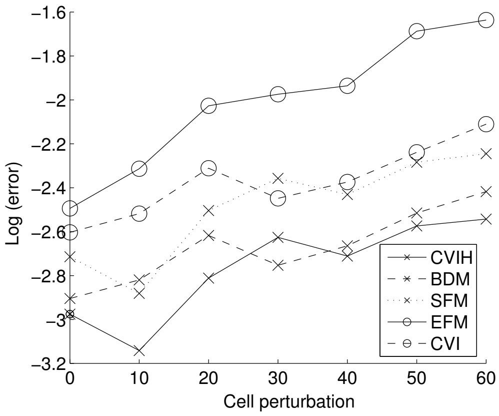

Figure 23. Comparing different methods. Base 10 logarithm of error.

For two degrees-of-freedom per edge in the two-dimensional case, the velocity should belong to the Brezzi–Douglas–Marini (BDM) space of order one, with the added condition of zero divergence [39, 40]. By the BDM method we will in the following mean streamlines traced by the above velocity field. See [41, 42, 30] for a comprehensive discussion of streamline tracing in 2D using low and high-order mixed finite-element velocity spaces.

In our final example, we compare all the methods introduced so far: BDM, CVIH, CVI, SFM, and EFM for the analytical solution in (58). We start with a 10×10 uniform partition of $[0, 1] \times [0, 1]$ and consider grids obtained by a $10i$ percent perturbation for $i = 1, \dots, 6$. Exact edge and half-edge fluxes will be computed for each grid. These fluxes will be used to construct interpolated velocity fields for all the methods. For each grid, streamlines are started from twenty random points within a random cell. This cell is selected randomly as one of three cells in the top row of the grid. The results are shown in Figure 23. Here, as also remarked for Case 4 in Section 5.1, SFM is better than EFM due to cancellation of errors. We also note that the CVIH and BDM methods are of approximately the same accuracy.

6. Summary and Concluding Remarks

This work has investigated streamline generation on irregular grids in 3D and has in particular focused on the problem of representing uniform flow on hexahedral grids. We have considered two standard methods, the standard flux-mapping method (SFM) [15, 47, 49] and the extended flux-mapping method (EFM) [28]. For irregular grids, these methods are based on a trilinear transformation of each grid cell to a unit cube together with a linear interpolation scaled by the Jacobian. The two methods only differ in the way they treat the Jacobian; see Eqs. (20) and (21). The major advantage of the SFM and EFM methods is that they both allow for fast and analytical integration of streamlines due to the linear flux interpolation. On the other hand, the methods share the same fundamental deficiency of being inaccurate for irregular (and rough) grids. That is, for irregular grids, the interpolated velocity field used in these methods will generally depend on the geometry of the grid cell, except for the 2D case, where a linear flux interpolation in reference space is sufficient to model the normal vectors $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$ exactly. In 3D, the flux of a uniform flow field varies quadratically for a general hexahedral cell, and also a flux interpolation cannot capture the variation of the normal vectors that describe the geometry of the cell.

To improve the tracing accuracy, we introduced a new method, which we called the corner-velocity interpolation (CVI) method. Instead of interpolating the velocity inside the cell based on discrete fluxes at cell edges, the method interpolates the velocity based on (reconstructed) velocities at the eight corner-points of a hexahedral cell. This way, we get a method that is generally less sensitive to the regularity of the cells and in particular is able to reproduce uniform flow regardless of the cell geometries, a property that is considered to be of high importance in solution of (elliptic) pressure equations. Moreover, the CVI method can easily be adapted to exploit the extra accuracy represented in half-edge fluxes, when these are available (see Section 5.6).

In Section 5, we compare the three tracing methods on a variety of simple test cases, focusing in particular on representing uniform flow in 3D. Our test cases in Sections 5.1 to 5.3 can be divided in two categories. In the first category, the irregularity of the grids diminish as they are refined. For these grids, all three methods converge, but CVI typically converges faster and is more accurate on each specific grid as the flow becomes uniform.

In the second category, we consider so-called *rough grids* and nonuniform refinements of these. (Nonuniform refinement may seem a bit strange on a first glance, but is in fact what will typically be used when refining a coarse simulation model towards an underlying geological model, which typically contains highly irregular cells used to model the complex structures of the underlying geology.) The test cases in this category reveal significant differences in the behavior of the three methods. For tests with uniform flow, the SFM and EFM methods do not only fail to reproduce uniform flow; they also fail to converge as the grids are (nonuniformly) refined. For nonuniform flow cases, our tests establish convergence of the CVI method, whereas the SFM and EFM methods either do not converge asymptotically or have a small convergence rate for the refinements levels observed.

The added accuracy of the CVI method comes at the cost of an increased computational complexity, which is imposed by the need to use numerical integration of a set of ODEs to compute streamlines. Since EFM and SFM use analytical integration, these methods will be significantly faster. We are therefore investigating various means to speed up the CVI method, for instance, integrating streamlines directly in physical space using an adaptive Euler predictor-corrector method [25]. The method proved to be quite successful in 2D, but has not yet been extended to 3D. Another tempting idea is to use a hybrid method, in which SFM is used for regular cells and CVI is used for irregular cells. This, however, requires some kind of error estimate for SFM and EFM and remains to be investigated.

Acknowledgments. The authors would like to thank Jan Nordbotten and two anonymous referees for useful comments and suggestions for improving the manuscript.

The research of K.–A. Lie and H. Hægland was funded by the Research Council of Norway under grants no. 158908/I30 and 173875/I30, respectively.

References

[1] I. Aavatsmark. An introduction to multipoint flux approxmations for quadrilateral grids. *Comput. Geosci.*, 6:405–432, 2002.

[2] K. Aziz and A. Settari. *Petroleum reservoir simulation*. Elsevier Applied Science Publishers, London, 1979.

[3] R.P. Batycky. *A three-dimensional two-phase field scale streamline simulator.*. PhD thesis, Stanford University, Dept. of Petroleum Engineering, 1997. http://www.streamsim.com/index.cfm?go=about.publications

[4] J. Bear. *Dynamics of fluids in porous media*. Environmental Science Series. American Elsevier Publishing Company, New York, 1972.

[5] J. Bear and A. Verruijt. *Modeling groundwater flow and pollution : with computer programs for sample cases*. Theory and applications of transport in porous media ; 2. D.Reidel, Dordrecht, 1987.

[6] J. Bensabat, Q. Zhou, and J. Bear. An adaptive pathline-based particle-tracking algorithm for the eulerian-lagrangian method. *Adv. Water Resour.*, 23:383–397, 2000.

[7] F. Bratvedt, K. Bratvedt, C.F. Buchholz, T. Gimse, H. Holden, L. Holden, and N.H. Risebro. Frontline and frontsim: two full scale, two-phase, black oil reservoir simulators based on front tracking. *Surveys Math. Indust.*, 3:185–215, 1993.

[8] F. Bratvedt, T. Gimse, and C. Tegnander. Streamline computations for porous media flow including gravity. *Transp. Porous Media*, 25(1):63–78, 1996.

[9] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer series in computational mathematics, 15. Springer-Verlag, New York, 1991.

[10] Z. Cai, J.E. Jones, S.F. McCormick, and T.F. Russell. Control-volume mixed finite element methods. *Comput. Geosci.*, 1:289–315, 1997.

[11] R.J. Charbeneau and R.L. Street. Modeling groundwater flow fields containing point singularities: Streamlines, travel times, and breakthrough curves. *Water Resour. Res.*, 15(6):1445–1450, 1979.

[12] H.-P. Cheng, J.-R. Cheng, and G.-T. Yeh. A particle tracking technique for the lagrangian-eulerian finite element method in multi-dimensions. *Int. J. Numer. Methods Engrg.*, 39:1115–1136, 1996.

[13] H. Cheng, I. Osako, A. Datta-Gupta, and M.J. King. A rigorous compressible streamline formulation for two- and three-phase black-oil simulation. Paper SPE 96866 in *Proceedings of the SPE Annual Technical Conference and Exhibition, Dallas, Texas*, 2005.

[14] P.G. Ciarlet. *The finite element method for elliptic problems*. North-Holland, Amsterdam, 1980.

[15] C. Cordes and W. Kinzelbach. Continous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements. *Water Resour. Res.*, 28(11):2903–2911, 1992.

[16] M. Crane, F. Bratvedt, K. Bratvedt, P. Childs, and R. Olufsen. A fully compositional streamline simulator. SPE 63156. In *Proceedings of the SPE Annual Technical Conference and Exhibition, Dallas, Texas, 1-4 Oct.*, volume 3, 2000.

[17] G. Di Donato and M.J. Blunt. Streamline-based dual-porosity simulation of reactive transport and flow in fractured reservoirs. *Water Resour. Res.*, 40(4), 2004.

[18] J.R. Dormand and P.J. Prince. A familiy of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, 6(1):19–26, 1980.

[19] L.J. Durlofsky. Accuracy of mixed and control volume finite element approximations to darcy velocity and related quantities. *Water Resour. Res.*, 30(4):965–973, 1994.

[20] M.G. Edwards. Unstructured, control-volume distributed full-tensor finite-volume schemes with flow based grids. *Comput. Geosci.*, 6:433–452, 2002.

[21] W. Gander and W. Gautschi. Adaptive quadrature – revisited. *BIT*, 40:84–101, 2000.

[22] R.H.J. Gmelig Meyling. A characteristic finite element method for solving non-linear convection-diffusion equations on locally refined grids. In *Proceedings of the second European Conference on the Mathmatics of Oil Recovery, Sept. 11-14, Arles, France*, 1990.

[23] R.H.J. Gmelig Meyling. Numerical methods for solving the nonlinear hyperbolic equations of porous media flow. In *Proceedings of the Third International Conference on Hyperbolic Problems, Uppsala, Sweeden*, 1990.

[24] D.J. Goode. Particle velocity interpolation in block-centered finite difference groundwater flow models. *Water Resour. Res.*, 26(5):925–940, 1990.

[25] H. Hægland, H.K. Dahle, I. Aavatsmark, G.T. Eigestad, and K.-A. Lie. Adaptive streamline tracing for streamline simulation on irregular grids. In *Proceedings of the XVI International Conference on Computational Methods in Water Resources, June, Denmark*, 2006.

[26] L. Ingebrigtsen, F. Bratvedt, and J. Berge. A streamline based approach to solution of three-phase flow. SPE 51904. In *SPE Reservoir Simulation Symposium, Houston, Texas, 14-17 February*, 1999.

[27] I. Javandel, C. Doughty, and C.-F. Tsang. *Groundwater transport : handbook of mathematical models.* Water resources monograph series ; 10. American Geophysical Union, Washington, 1984.

[28] E. Jimenez, K. Sabir, A. Datta-Gupta, and M.J. King. Spatial error and convergence in streamline simulation. SPE 92873. In *SPE Reservoir Simulation Symposium, Woodlands, Texas, 31 Jan. -2 Feb.*, 2005.

[29] R. Juanes and K.-A. Lie. A front-tracking method for efficient simulation of miscible gas injection processes. SPE 93298. In *SPE Reservoir Simulation Symposium, Woodlands, Texas, 31 Jan. -2 Feb.*, 2005..

[30] R. Juanes and S.F. Matringe. Unified formulation of velocity fields for streamline tracing on two-dimensional unstructured grids. *Comput. Methods. Appl. Mech. Engrg.*, submitted 2006.

[31] E.F. Kaasschieter. Mixed finite elements for accurate particle tracking in saturated groundwater flow. *Adv. Water Resour.*, 18(5):277–294, 1995.

[32] M.J. King and A. Datta-Gupta. Streamline simulation: A current perspective. *In Situ*, 22(1):91–140, 1998.

[33] P. Kipfer, F. Reck, and G. Greiner. Local exact particle tracing on unstructured grids. *Comput. Graph. Forum*, 22(2):1–9, 2003.

[34] R.A. Klausen and R. Winther. Robust convergence of multi point flux approximation on rough grids. *Numer. Math.*, 2006. To appear.

[35] D. Knight and G. Mallinson. Visualizing unstructured flow data using dual stream functions. *IEEE Trans. Vis. Comput. Graph.*, 2(4):355–363, 1996.

[36] L.P. Lebedev and M.J. Cloud. *Tensor analysis.* World Scientific, River Edge, N.J., 2003.

[37] K.-A. Lie and R. Juanes. A front-tracking method for the simulation of three-phase flow in porous media. *Comput. Geosci.*, 9(1):29–59, 2005.

[38] N. Lu. A semianalytical method of path line computation for transient finite-difference groundwater flow models. *Water Resour. Res.*, 30(8):2449–2459, 1994.

[39] S.F. Matringe. Accurate streamline tracing and coverage. Master's thesis, Stanford university, Dept. of petr. eng., 2004.

[40] S.F. Matringe and M.G. Gerritsen. On accurate tracing of streamlines. In *Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, Texas, 26-29 Sep.*, 2004.

[41] S.F. Matringe, R. Juanes, and H.A. Tchelepi. Streamline tracing on general triangular and quadrilateral grids. Paper SPE 96411 in *Proceedings of the SPE Annual Technical Conference and Exhibition, Dallas, Texas*, 2005.

[42] S.F. Matringe, R. Juanes, and H.A. Tchelepi. Robust streamline tracing for the simulation of porous media flow on general triangular and quadrilateral grids. *J. Comput. Phys.*, submitted 2006.

[43] R.L. Naff, T.F. Russell, and J.D. Wilson. Shape functions for velocity interpolation in general hexahedral cells. *Comput. Geosci.*, 6(3-4):285–314, 2002.

[44] J.R. Natvig, K.-A. Lie, B. Eikemo, and I. Berre. A discontinuous Galerkin method for computing single-phase flow in porous media. Submitted 2006.

[45] A. Oliveira and A.M. Baptista. On the role of tracking on eulerian-lagrangian solutions of the transport equation. *Adv. Water Resour.*, 21:539–554, 1998.

[46] D. Pokrajec and R. Lazic. An efficient algorithm for high accuracy particle tracking in finite elements. *Adv. Water Resour.*, 25:353–369, 2002.

[47] D.W. Pollock. Semi-analytical computation of path lines for finite-difference models. *Ground Water*, 26(6):743–750, 1988.

[48] M. Prévost. *Accurate coarse reservoir modeling using unstructured grids, flow-based upscaling and streamline simulation.* PhD thesis, University of Stanford, 2003. http://geothermal.stanford.edu/pereports/search.htm.

[49] M. Prévost, M. G. Edwards, and M.J. Blunt. Streamline tracing on curvilinear structured and unstructured grids. *SPE J.*, pages 139–148, June 2002.

[50] T.F. Russell and R.W Healy. Analytical tracking along streamlines in temporally linear raviart-thomas velocity fields. In *Proceedings of the XIII International Conference on Computational Methods in Water Resources, 25-29 June, Alberta, Canada*, 2000.

[51] I.A. Sadarjoen, T. van Walsum, A.J.S. Hin, and F.H. Post. Particle tracing algorithms for 3d curvilinear grids. In G.M. Nielson, H. Hagen, and H. Müller, editors, *Scientific visualization: Overviews, methodologies, and techniques*, chapter 14, pages 311–335. IEEE Computer Society, Los Alamitos, Calif., 1997.

[52] A.L. Schafer-Perini and J.L. Wilson. Efficient and accurate front tracking for two-dimensional groundwater flow models. *Water Resour. Res.*, 27(7):1471–1485, 1991.

[53] J.M. Shafer. Reverse pathline calculation of time-related capture zones in nonuniform flow. *Ground Water*, 25(3):283–289, 1987.

[54] L.F. Shampine. Interpolation for Runge-Kutta methods. *SIAM J. Numer. Anal.*, 22(5):1014–1027, 1985.

[55] L.F. Shampine and M.W. Reichelt. The MATLAB ODE suite. *SIAM J. Sci. Comput.*, 18(1):1–22, 1997.

[56] S. Shirayama. Processing of computed vector fields for visualization. *J. Comput. Phys.*, 106:30–41, 1993.

[57] T. Strid, A. Rizzi, and J. Oppelstrup. Development and use of some flow visualization algorithms. *Lecture series/Von Karman Institute for Fluid Dynamics*, (7):1–56, 1989.

[58] M.R. Thiele. Streamline simulation. In *Proceedings of the Sixth International Forum on Reservoir Simulation, Fuschl, Austria*, 2001.

[59] M.R. Thiele, R.P. Batycky, and M.J. Blunt. A streamline-based 3D field-scale compositional reservoir simulator, SPE 38889. In *Proceedings of the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 5-8 Oct.*, 1997.

[60] M.R. Thiele, R.P. Batycky, M. Iding, and M. Blunt. Extension of streamline-based dual porosity flow simulation to realistic geology. In *Proceedings of the 9th European Conference on the Mathmatics of Oil Recovery, Cannes, France*, 2004.

[61] M.R. Thiele, R.P. Batycky, and L.K. Thomas. Miscible WAG simulations using streamlines. In *Proceedings of the 8th European Conference on the Mathmatics of Oil Recovery, Freiberg, Germany*, 2002.

[62] S. Verma and K. Aziz. A control-volume scheme for flexible grids in reservoir simulation, SPE 37999. In *SPE Reservoir Simulation Symposium, Dallas, Texas, 8-11 June*, 1997.

[63] C. Zheng and G.D. Bennett. *Applied contaminant transport modeling.* Wiley-Interscience, New York, second edition, 2002.