

Streamline tracing on irregular geometries

Atgeirr Flø Rasmussen
SINTEF ICT

July 15, 2010

Abstract

The accurate and efficient tracing of streamlines is fundamental to any streamline-based simulation method.

For grids with irregular cell geometries, such as corner-point grids with faults or Voronoi-diagram (pebi) grids, most efforts to trace streamlines have been focused on subdividing irregular cells into sets of simpler subcells, typically hexahedra or simplices. Then one proceeds by reconstructing the velocity field on, and tracing through, the sets by a more basic algorithm.

One such basic algorithm applies to incompressible flow on simplices. In that case there is a cell-wise constant velocity that is consistent with given face fluxes, as long as those face fluxes sum to zero for the cell. We give an efficient and simple formulation of this algorithm using barycentric coordinates.

Another approach to irregular cell tracing is computing the streamline directly on the complex cell geometry. We give a new method based on generalized barycentric coordinates for direct tracing on arbitrary convex polygons, which generalizes the corner velocity interpolation method of Hægland et al (2007). The method generalizes to convex polytopes in 3D, with a restriction on the polytope topology near corners that is shown to be satisfied by several popular grid types.

Introduction

Streamline methods for reservoir simulation have been studied extensively. For a comprehensive overview see the book by Datta-Gupta and King (2007). Streamline tracing is a critical part of any such method, and consists of solving the streamline ODE on some computational grid:

$$\frac{d\mathbf{x}}{d\tau} = \mathbf{v}(\mathbf{x}), \quad (1)$$

where $\mathbf{x} = (x^1, \dots, x^n)$ is a point in space and \mathbf{v} is the velocity field. The parameter τ is known as the streamline “time-of-flight”. Throughout the paper we will use boldface to signify vector quantities.

For our purposes the velocity field is considered fixed in time. Consequently, the streamlines will not correspond to path lines if the actual velocity field \mathbf{v} evolves in time. Typically, a flow solver provides a discrete velocity or flux field, and we replace the true velocity field \mathbf{v} in (1) by some approximation $\hat{\mathbf{v}}$. This is typically accomplished by means of a velocity interpolation technique. In our case, we assume that a discrete flux field is given in terms of an oriented normal flux on each grid face. For simplicity, we will assume homogenous porosity per grid cell, which simply scales the resulting cell time-of-flight, and leave it out of the equations.

In this paper we first describe Pollock’s method as an example, then we describe a new formulation for incompressible streamline tracing on simplicial grids. Then we describe the extended corner velocity interpolation method, which allows tracing on almost arbitrary convex cells, and give some numerical examples.

Pollock’s method

As an example of a streamline method, we give a short description of the semi-analytical method by Pollock (1988). For simplicity we describe the 2D variant. It is designed for structured Cartesian grids, but variants by Cordes and Kinzelbach (1992), Prevost et al. (2002) and Jimenez et al. (2005) are often used for grids with general hexahedral cells. In Pollock’s method, the velocity field interpolation is given on the unit square by linear interpolation in each component separately. Thus, we replace the velocity field \mathbf{v} in (1) by the linear approximation

$$\hat{\mathbf{v}}(\mathbf{x}) = \begin{pmatrix} -(1-x^1)f_1 + x^1f_2 \\ -(1-x^2)f_3 + x^2f_4 \end{pmatrix} = \begin{pmatrix} (f_1 + f_2)x^1 - f_1 \\ (f_3 + f_4)x^2 - f_3 \end{pmatrix} \quad (2)$$

where f_1, \dots, f_4 denote the outward normal velocities at the left, right, bottom and top boundaries, respectively. Because the first velocity component depends only on x^1 and the second only on x^2 , the components of the streamline ODE (1) can be solved independently. The analytic solution is given by

$$x^i(\tau) = \left(x^i(0) - \frac{f_{2i-1}}{f_{2i-1} + f_{2i}} \right) \exp((f_{2i-1} + f_{2i})\tau) - \frac{f_{2i-1}}{f_{2i-1} + f_{2i}}. \quad (3)$$

For other methods, the interpolation of $\hat{\mathbf{v}}$ may not allow the computation of analytical solutions. In that case, we need to solve (1) by numerical means. This is the case for the extended CVI method described in this paper.

Streamline tracing on simplicial grids

For the special case of incompressible flow on simplicial grids, that is, triangulations in 2D and tetrahedral grids in 3D, there exists a cell-wise constant velocity field that is consistent with the face fluxes. In n dimensions a constant velocity has n degrees of freedom, and on a simplex we have $n + 1$ faces at which we need to satisfy consistency. However, incompressibility implies that the fluxes sum to zero (away from wells), which reduces the number of independent consistency equations by one.

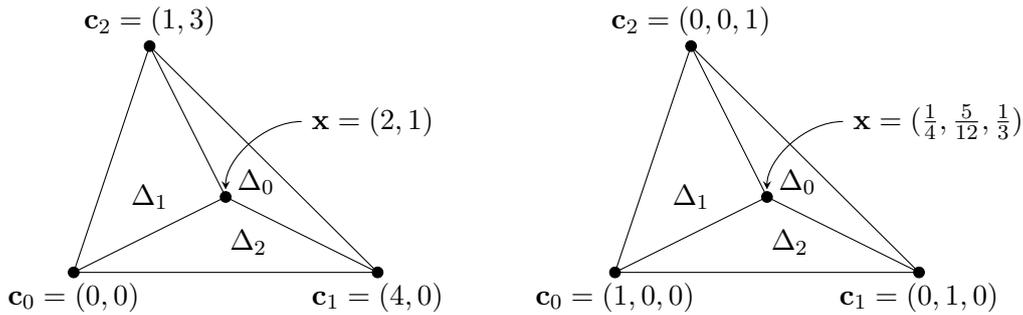


Figure 1 Cartesian vs. barycentric coordinates.

Note: One may also arrive at a cell-wise constant velocity field if solving for pressure with P_1 piecewise linear elements, and taking the cell-wise gradient of the resulting pressure field. This velocity field will not in general have a continuous normal component across cell boundaries, and is unsuited for streamline tracing. Even in the incompressible case the cell fluxes will not in general sum to zero.

When the face fluxes do sum to zero, for example if computed by a finite difference or finite volume method, there is an elegant formulation using barycentric coordinates. First, we review some properties of such coordinates.

For any simplex S with corners $\{c_0, \dots, c_n\}$ in \mathbf{R}^n , the barycentric coordinate functions b_0, \dots, b_n are defined by

$$b_i(\mathbf{x}) = \frac{\Delta_i(\mathbf{x})}{\sum_{i=0}^n \Delta_i(\mathbf{x})}.$$

Here $\Delta_i(\mathbf{x})$ is the signed volume (or area) of the simplex with the corner c_i replaced by \mathbf{x} . See Figure 1 for an example. They have the following properties:

1. $\sum_{i=0}^n b_i(\mathbf{x}) = 1$.
2. $b_i(c_j) = \delta_{ij}$.
3. $b_i(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in S$.
4. $\sum_{i=0}^n b_i(\mathbf{x})c_i = \mathbf{x}$.
5. The $\{b_i\}$ are linear.

As a direct consequence of property 1 above, vectors written in barycentric coordinates will sum to zero, since a vector is the difference between two points. Assume that for the simplex S we are given a set of outward normal fluxes $\{f_i\}$, where any particular f_i is the normal flux through the subsimplex (face in 3D or edge in 2D) S_i , which is opposite to c_i . If the flow is incompressible, the fluxes sum to zero, and there is a constant velocity field consistent with the normal fluxes.

The normal flux of a velocity field \mathbf{v} over a subsimplex S_i is given by $\int_{S_i} \mathbf{v} \cdot \mathbf{n}_i dx$. For a constant velocity field this is equal to the subsimplex area or length A_i multiplied by the normal component of the velocity vector. We will prove the following assertion:

Lemma 1 For a set of normal fluxes $\mathbf{f} = (f_i)$ on $S \subset \mathbf{R}^n$ with the property that $\sum_{i=0}^n f_i = 0$, the constant velocity

$$\hat{\mathbf{v}} = -\mathbf{f}/(nV) \tag{4}$$

in barycentric coordinates is consistent with the normal fluxes \mathbf{f} . In the above, V is the volume of S .

Proof. In Cartesian coordinates,

$$\hat{\mathbf{v}} = -\frac{1}{nV} \sum_{i=0}^n f_i \mathbf{c}_i.$$

Without loss of generality, we show that $\hat{\mathbf{v}}$ is consistent with f_0 . If f_0 is zero, we substitute $f_1 = -\sum_{i=2}^n f_i$ and obtain

$$\hat{\mathbf{v}} = -\frac{1}{nV} \sum_{i=2}^n f_i (\mathbf{c}_i - \mathbf{c}_1).$$

Since all the vectors $\mathbf{c}_i - \mathbf{c}_1$ are tangent to S_0 , clearly the normal component of $\hat{\mathbf{v}}$ and therefore the normal flux through S_0 is zero.

If f_0 is nonzero, we write

$$\hat{\mathbf{v}} = -\frac{f_0}{nV} \left(\mathbf{c}_0 + \sum_{i=1}^n \frac{f_i}{f_0} \mathbf{c}_i \right) = -\frac{f_0}{nV} (\mathbf{c}_0 - \mathbf{r}).$$

Since $-\sum_{i=1}^n f_i/f_0 = 1$, we know that \mathbf{r} must lie in the hyperplane tangent to S_0 . Therefore the outward pointing normal component of $\mathbf{c}_0 - \mathbf{r}$ must equal $-d_0$, where d_0 is the distance from the hyperplane to \mathbf{c}_0 . Since $d_0 A_0 = nV$, it follows that the normal flux is equal to $-f_0(-d_0)A_0/(nV) = f_0$. \square

Tracing a streamline through S now may be done by following this procedure:

1. Find the barycentric coordinates $\mathbf{x}_0 = \mathbf{x}(\tau_0)$ of the entry point with respect to S . If the entry point is given in barycentric coordinates for the intercell boundary subsimplex (a triangle in 3D, a line in 2D), this step consists of a simple shuffling of coordinates.
2. Compute the velocity $\hat{\mathbf{v}}$ in barycentric coordinates using equation (4). That is, each component is given by $\hat{v}_i = -f_i/(nV)$.
3. Equation (1) has the solution $\mathbf{x}(\tau) = \mathbf{x}_0 + (\tau - \tau_0) \cdot \hat{\mathbf{v}}$. Here τ_0 is the entry time-of-flight. Find the exit point and time-of-flight as follows:
 - (a) For all i such that $\hat{v}_i < 0$, compute the intersection times $\tau_i = -x_i(\tau_0)/\hat{v}_i$. For all other i , set $\tau_i = -\infty$.
 - (b) The minimum non-negative intersection time τ_m is the time-of-flight through S . The index m at which this minimum is attained indicates the boundary subsimplex that was hit. If there are multiple candidates, choose the one for which the corresponding subsimplex outwards flux is greatest.
 - (c) The exit point is given in barycentric coordinates with respect to S by $\mathbf{x}(\tau_m) = \mathbf{x}_0 + \tau_m \cdot \hat{\mathbf{v}}$.
4. Transform $\mathbf{x}(\tau_m)$ to barycentric coordinates on the intercell boundary subsimplex by removing the coordinate for which $\tau_m = \tau_i$ (which must be zero).
5. S now becomes the neighbouring simplex that is adjacent through subsimplex m . Repeat the procedure.

The above procedure is robust. It cannot be trapped on an edge or vertex, and will always exit S through an outflow face. However when using floating point numerics, it is prudent to add a rescaling of the coordinates after step 4, to ensure that they sum to one.

Extending the CVI method to arbitrary convex polytopes

In Hægland et al. (2007), the authors describe *corner velocity interpolation* (CVI), a streamline tracing method aiming to produce streamlines that are more accurate than those produced by the methods of Pollock (1988), Cordes and Kinzelbach (1992), Prevost et al. (2002) and Jimenez et al. (2005). Our goal is slightly different, namely to trace streamlines on arbitrary convex polytopes without resorting to subdivision of the polytope. We describe an extension to the CVI method that achieves this on arbitrary convex polygons in 2D and on a large subset of 3D polytopes.

This is closely related to the issue of velocity field interpolation in general, which is also of interest in the mixed finite element setting. In Klausen and Stephansen (2010) this aspect of the method is described.

The basic idea is to use *generalized barycentric coordinates* to define an interpolation for the velocity field. Assume that a convex polytope $E \subset \mathbf{R}^n$ is given, which is bounded by hyperplanes. In particular, curved boundaries are disallowed. Let the corners of E be $\{\mathbf{c}_1, \dots, \mathbf{c}_k\}$. Then a set of functions $\{\lambda_1, \dots, \lambda_k\}$ are generalized barycentric coordinates for E if the following hold:

1. $\sum_{i=1}^k \lambda_i(\mathbf{x}) = 1$.
2. $\lambda_i(\mathbf{c}_j) = \delta_{ij}$.
3. $\lambda_i(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in E$.
4. $\sum_{i=1}^k \lambda_i(\mathbf{x})\mathbf{c}_i = \mathbf{x}$.

Two important examples of generalized barycentric coordinate functions are Wachspress' coordinates described in Wachspress (1975), Meyer et al. (2002) and the Mean Value coordinates of Floater (2003). We do not give any details of the computation of $\lambda_i(\mathbf{x})$ here, but refer the reader to the above-mentioned papers. Both families produce smooth functions on the polytope E . While being initially described for the 2D case, both have generalizations to 3D. The Wachspress coordinates coincide with the inverse bilinear mapping for quads, so when using those coordinates the extended CVI method reduces to the regular CVI method for quad grids.

Given such a set of functions, we define a velocity interpolation on E by

$$\hat{\mathbf{v}}(\mathbf{x}) = \sum_{i \in C(E)} \mathbf{v}_i \lambda_i(\mathbf{x}), \quad (5)$$

where $\{\mathbf{v}_i\}$ are the corner velocities at the corners of E , denoted by $C(E)$.

The extended CVI method consists of the following steps:

1. For each cell element E , pre-compute corner velocities \mathbf{v}_i consistent with a given flux field.
2. Upon a streamline reaching a cell boundary entry point, integrate it with the velocity function given by (5) until reaching the cell boundary again. That is the exit point, which will be the entry point for the next segment of the streamline, traced through the neighbouring cell.

We now give some detail of the two steps above.

Finding corner velocities

In order to compute the corner velocities in step 1 of the method we must satisfy the consistency requirement

$$\int_{E_j} \mathbf{v}_i \cdot \mathbf{n} \, dx = f_j.$$

In the above, E_j is the j 'th face of E , \mathbf{v}_i is the corner velocity at the i 'th corner of E and f_j is the outwards oriented flux through E_j . For any given \mathbf{v}_i , the equation must be satisfied for all adjacent E_j . For planar faces this reduces to

$$\mathbf{n}_j \cdot \mathbf{v}_i = \frac{f_j}{A_j}, \quad (6)$$

where \mathbf{n}_j is the unit outer normal of E_j , and A_j is its area (or length).

In 2D this equation is always solvable as long as the cell E is strictly convex. Each corner has two adjacent edges, and the normal vectors are linearly independent due to strict convexity.

In 3D the equation may not always have a solution, even for strictly convex E . However, many types of grids do have the property that each corner of a cell has three adjacent faces. Example cell shapes include hexahedra, simplices, arbitrary prisms and even dodecahedra. On the other hand pyramids (one corner has four), octahedra (all corners have four) and icosahedra (all corners have five) do not. Pebi grids in 3D and 2.5D (extruded) will in general work well with the method. We prove the assertion for 3D Voronoi diagrams under a uniqueness assumption.

Lemma 2 *If there are no five cospherical Voronoi sites, corresponding to uniqueness of the dual Delaunay tetrahedral grid, each Voronoi cell corner will have exactly three adjacent faces.*

Proof. Let E be the cell generated by the Voronoi site \mathbf{x}_E . Assume that there is a corner with four or more adjacent faces. On each such face, all its points, including the corner itself, are equidistant between \mathbf{x}_E and one other site. So the corner must be equidistant to \mathbf{x}_E and at least four more sites. Then those (at least) five sites are cospherical, which contradicts our assumption. \square

Integrating the velocity field

In order to actually trace a streamline through E , we must use some numerical ODE solver, and this is the primary drawback of the extended CVI method, compared to semi-analytic methods.

Assuming that tracing starts at a point \mathbf{x}_0 on some inflow boundary face of the cell, we must trace forwards by solving (1) numerically. A simple scheme like Euler's method may or may not be sufficient, depending on accuracy requirements and the application. We have used the second-order Heun's method, which is an explicit second-order method also known as the Euler-Trapezoidal predictor-corrector method. The scheme is given by

$$\begin{aligned} \mathbf{x}_{n+1}^p &= \mathbf{x}_n + h\hat{\mathbf{v}}(\mathbf{x}_n) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \frac{h}{2} (\hat{\mathbf{v}}(\mathbf{x}_n) + \hat{\mathbf{v}}(\mathbf{x}_{n+1}^p)) \end{aligned}$$

Choosing the step-size h can be challenging. An initial step-size may be chosen proportional to some characteristic length, for instance the square root (cube root in 3D) of the cell volume, divided by some characteristic velocity, the simplest approach using a cell average velocity.

The difference between \mathbf{x}_{n+1}^p and \mathbf{x}_{n+1} may be used to control the error, exploiting the fact that Heun's method is a second-order method with an embedded first-order method. Similar embedded method pairs of higher degree such as the Runge-Kutta-Fehlberg 4(5) pair or the Dormand-Prince 5(4) pair may be used for increased accuracy.

Another challenge is detecting the point at which the streamline exits the cell. Given the generalized barycentric coordinates $\{\lambda_i\}$ for a point it is easy to check, since a point lies inside the cell if and only if all coordinates are nonnegative. We need to compute these coordinates for the purpose of calculating $\hat{\mathbf{v}}$, so there is no extra computational cost.

Once we know that the integration has passed outside the cell, we need to find the point of intersection with the cell boundary. A simple approach is to use a function ϕ that is defined on the line between \mathbf{x}_n and \mathbf{x}_{n+1} . The indicator function ϕ should be negative on the outside and positive on the inside of E . For example one may use $\phi(\mathbf{x}) = \min_i \lambda_i(\mathbf{x})$. Then one can use a nonlinear equation solver for bracketed zeros to find the intersection, for example one of the modified regula falsi solvers described by Ford (1995) or the method of Brent (1973). The regula falsi variants are easier to implement correctly, but Brent's method may give faster convergence. This approach only gives first order accuracy at the intersection point, however. If higher accuracy is needed one should use a continuous Runge-Kutta method to define the curve between \mathbf{x}_n and \mathbf{x}_{n+1} along which the zero of ϕ is sought, see for example Owren and Zennaro (1992). For order up to three, Hermite interpolation of the solution curve will yield sufficient accuracy.

Consistency and uniform flow

By construction, the corner velocities are consistent with the given discrete face fluxes. This in itself does not guarantee that the interpolated velocity field is in $H(\text{div})$, that is, the normal components are continuous across cell boundaries.

Lemma 3 *The extended CVI method produces a velocity field in $H(\text{div})$.*

Proof. We use the linear precision property (property 4) of the generalized barycentric coordinates:

$$\sum_{i \in C(E)} \lambda_i(\mathbf{x}) \mathbf{c}_i = \mathbf{x}.$$

Let $\mathbf{x} \in E_j$ and let $C(E_j)$ be the set of corners of the face E_j . Then we have that

$$\mathbf{x} = \sum_{i \in C(E_j)} \lambda_i(\mathbf{x}) \mathbf{c}_i + \sum_{i \in C(E) \setminus C(E_j)} \lambda_i(\mathbf{x}) \mathbf{c}_i.$$

By convexity of E , all \mathbf{c}_i in the second sum lie strictly on one side of E_j . The corresponding λ_i cannot be negative by property 3 and must therefore all be zero. So only the $\lambda_i : i \in C(E_j)$ are positive. Since they sum to one by property 1, and the normal component for every $\mathbf{v}_i : i \in C(E_j)$ is equal to f_j/A_j , the normal component of $\hat{\mathbf{v}}$ must be the same constant for all $\mathbf{x} \in E_j$.

The same analysis holds for the cell F for which E_j is the common face, therefore the normal components are the same on both sides of the face. \square

Preserving uniform flow is considered to be an important feature for a streamline tracing algorithm. The lack of this feature often implies that as the grid is refined and the accuracy of the discrete flux field increases, the streamlines do not converge to the true solution.

Lemma 4 *The extended CVI method preserves uniform flow. Given a constant velocity field \mathbf{v} , the velocity interpolation $\hat{\mathbf{v}}(\mathbf{x})$ defined by (5) and (6) will be the same at all points: $\hat{\mathbf{v}}(\mathbf{x}) \equiv \mathbf{v}$.*

Proof. For any cell face E_j , the \mathbf{v} satisfies (6), since the flux through E_j is given by

$$f_j = (\mathbf{v} \cdot \mathbf{n}_j) A_j.$$

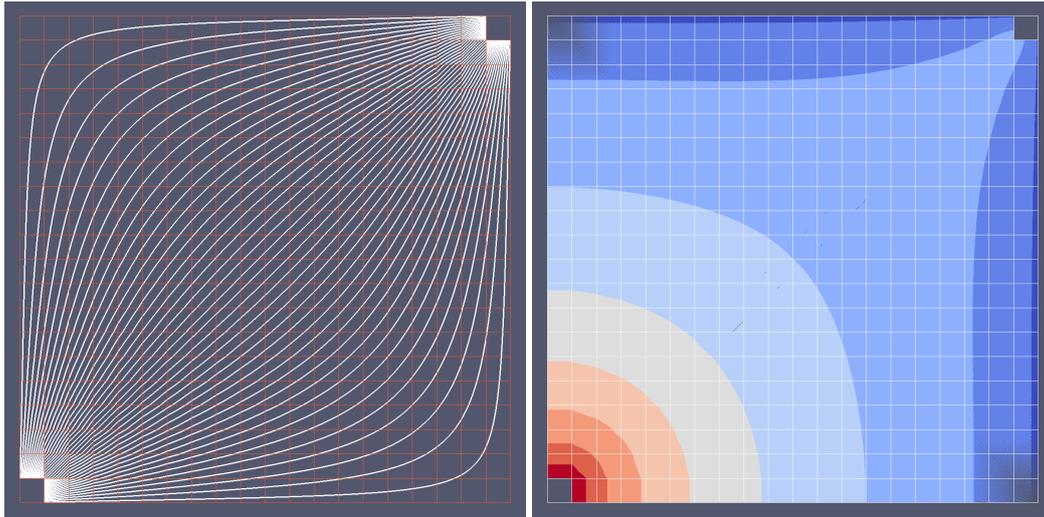


Figure 2 Streamlines for homogenous quarter-five-spot case with 20×20 cells. Left: Sparse streamlines. Right: Dense streamlines that have been coloured according to logarithmic time-of-flight with a discrete set of colours to show isocontours.

Therefore all the \mathbf{v}_i are equal to \mathbf{v} , and

$$\hat{\mathbf{v}}(\mathbf{x}) = \sum_{i \in C(E)} \mathbf{v}_i \lambda_i(\mathbf{x}) = \mathbf{v} \sum_{i \in C(E)} \lambda_i(\mathbf{x}) = \mathbf{v},$$

where the last equality follows from property 1 of the generalized barycentric coordinates $\{\lambda_i\}$. □

Numerical examples

Figure 2 shows the performance of extended CVI streamlines applied to a Cartesian grid. The implementation is using Wachspress Coordinates, so for this case the extended CVI method reduces to the regular one. When tracing from a source to a sink, we choose to terminate the streamlines on the source/sink cell boundaries. This is a simple approach, but may not yield sufficiently accurate time-of-flight, especially on coarse grids with many wells. For such cases a more accurate model of the near-well region is needed, we do not consider this issue within the scope of this paper.

Figure 3 shows the performance of extended CVI streamlines applied to a polygonal grid. As can be inferred from the left figure, the method reproduces uniform flow. The right figure shows that the produced streamlines are smooth and quite robust to grid effects. Near the source and sink there is some distortion, but that can mostly be attributed to the coarse spatial resolution and its effect on the accuracy of the flow solver.

Figure 4 shows that the computed time-of-flight contours are quite smooth, even for the coarser grid. Close to the injection cell, as shown in Figure 5, it can be seen that the contours are distorted into the shape of the injector, but then become smoother as they advance.

Conclusions and future work

We have described a simple and efficient formulation of basic incompressible streamline tracing on simplices. A direction worth investigating would be looking for similar, efficient formulations for streamline tracing in compressible flow.

A new method has been described for direct streamline tracing on polygon and polytope cells. The method extends the CVI method of Hægland et al. (2007). Further testing of the method will compare its performance to existing methods based on subdivision, especially in 3D.

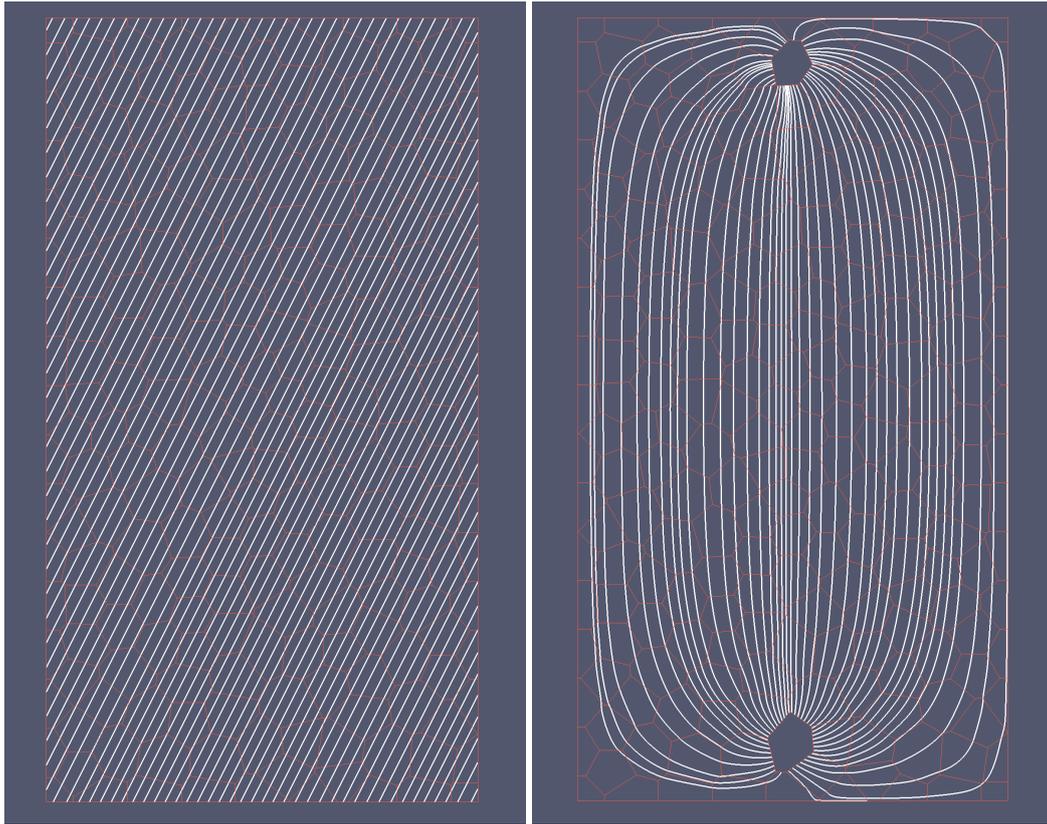


Figure 3 Sparse streamlines for a homogenous polygonal grid case with 232 cells. Left: uniform flow. Right: streamlines traced from injector cell to producer cell.

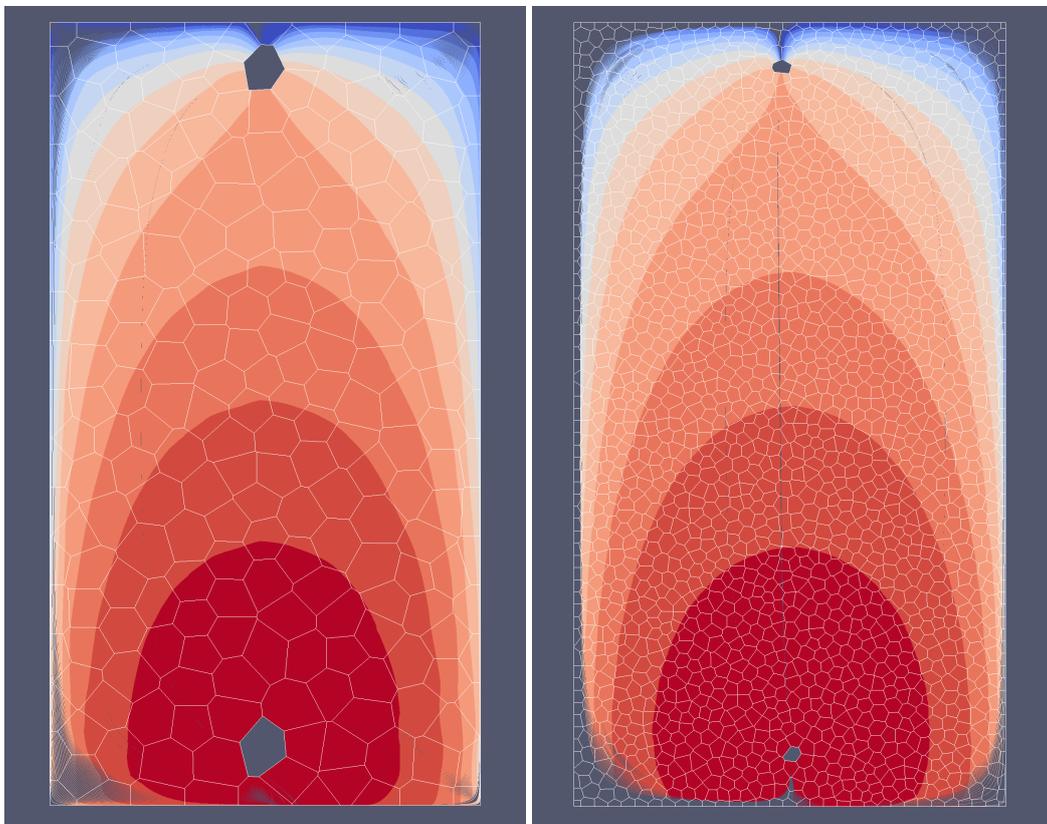


Figure 4 Dense streamlines for injection in homogenous polygonal grids. Streamlines have been coloured according to time-of-flight with a discrete set of colours to show isocontours. Left: 232 cells. Right: 2079 cells.

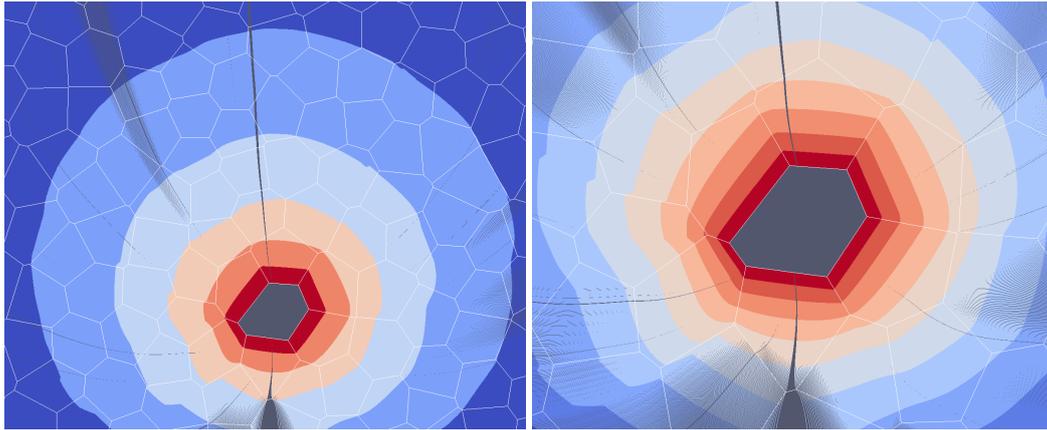


Figure 5 Closeups near source of the 2079 cell case. Coloured according to logarithmic time-of-flight.

Acknowledgments

I would like to thank Runhild Aae Klausen and Anette Stephansen for ideas and discussion about the extended CVI method.

I would also like to thank my colleagues Knut-Andreas Lie, Jostein Natvig, Halvor Møll Nilsen and Bård Skaflestad for useful discussions about streamline methods, collaboration on implementation and for creating test case grids and data files.

The research was partially funded by the Research Council of Norway under grant no. 186935.

References

- Brent, R.P. [1973] *Algorithms for Minimization without Derivatives*. Prentice-Hall.
- Cordes, C. and Kinzelbach, W. [1992] Continuous groundwater velocity fields and path lines in linear, bilinear and trilinear finite elements. *Water Resources Research*, **28**(11), 2903–2911.
- Datta-Gupta, A. and King, M.J. [2007] *Streamline Simulation: Theory and Practice*, vol. 11 of *SPE Textbook Series*. Society of Petroleum Engineers.
- Floater, M.S. [2003] Mean value coordinates. *Comp. Aided Geom. Design*, **20**, 19–27.
- Ford, J.A. [1995] Improved algorithms of illinois-type for the numerical solution of nonlinear equations. Tech. Rep. CSM-257, University of Essex.
- Hægland, H., Dahle, H.K., Eigestad, G.T., Lie, K.A. and Aavatsmark, I. [2007] Improved streamlines and time-of-flight for streamline simulation on irregular grids. *Adv. Water Resour.*, **30**(4), 1027–1045, doi: 10.1016/j.advwatres.2006.09.00.
- Jimenez, E., Sabir, K., Datta-Gupta, A. and King, M. [2005] Spatial error and convergence in streamline simulation. *SPE Reservoir Simulation Symposium, Houston, TX, January 31-February 2, 2005*, SPE 92873.
- Klausen, R.A. and Stephansen, A.F. [2010] On rough grids, convergence and reproduction of uniform flow. *Submitted for ECMOR XII*, Oxford, UK.
- Meyer, M., Barr, A., Lee, H. and Desbrun, M. [2002] Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools*, **7**(1), 13–22.
- Owren, B. and Zennaro, M. [1992] Derivation of efficient continuous explicit runge-kutta methods. *SIAM J. Sci. Stat. Comput.*, **13**, 1488–1501.
- Pollock, D. [1988] Semi-analytical computation of path lines for finite-difference models. *Ground Water*, **26**(6), 743–750.
- Prevost, M., Edwards, M.G. and Blunt, M.J. [June 2002] Streamline tracing on curvilinear structures and unstructured grids. *SPE Journal*, 139–148.
- Wachspress, E.L. [1975] A rational finite element basis.