

Streamline Tracing on Irregular Grids

Cand. Scient Thesis in Computational Science

Håkon Hægland

Department of Mathematics
University of Bergen



December 1, 2003

Preface

The work with this thesis has been an inspiring process. Although the progress with it has been periodically slow, I have learned a lot, and I have enjoyed working with it.

First, I would like to thank my supervisors Helge K. Dahle, Ivar Aavatsmark and Magne S. Espedal for a most interesting assignment and continuous assistance. Especially, I will thank Helge for feedback during the last few weeks.

I would also thank Torbjørn Aadland and Jarle Haukås for support using the MPFA-solver and for using their source code. Thanks to Jan Nordbotten for help on the MPFA-algorithm and for validating the source code, and to Geir Terje Eigestad for help on different subjects. I would also like to thank Tom Russell for assistance with the Piola transformation.

The streamline tracing method described in this thesis is to be implemented in 3D, for two phase flow problems. The 3D part is almost completed, but due to the time limit, 3D-results is not included.

Contents

Preface	i
1 Introduction	1
2 Basic concepts	3
2.1 Flow through porous media	3
2.2 Reservoir simulation	3
3 Flow and transport equations	5
3.1 The IMPES Equations	5
3.2 Numerical solution: MPFA	7
4 Streamlines	8
4.1 Eulerian and Lagrangian representations of fluid motion	8
4.2 Pathlines and Streamlines	8
4.3 The stream function	10
5 Irregular Grids	12
5.1 Structured logically Cartesian grids	12
6 The Bilinear Transformation	16
6.1 The transformation	16
6.2 Properties	17
6.2.1 Mapping of straight lines	17
6.2.2 The midpoint	18
6.2.3 The Jacobi matrix and the Jacobi determinant	18
6.3 The inverse transformation	19
6.4 Flux calculation	19
7 The Time-of-Flight	22
7.1 Time-of-Flight	22

8 Streamline Tracing	23
8.1 Streamline Tracing	23
8.2 Pollock Tracing in 2D	24
8.3 Invariance of streamlines w.r.t. a scale of velocities	26
8.4 Flux mappings	27
8.5 Velocity mappings	29
8.6 A consistent tracing velocity	31
8.6.1 The Prévost derivation.	31
8.7 The Prévost method	34
8.8 Direct transformation of the flow equation	34
8.9 Tracing on irregular grids	35
9 The Piola Transformation	36
9.1 The inverse transformation	36
9.2 Flux conservation	36
9.3 Reproduction of uniform flow	37
10 Results	39
10.1 Uniform velocity field	39
10.1.1 Shape of streamlines	41
10.1.2 Time-of-flight	41
10.2 Circular velocity field	42
10.2.1 Shape of streamlines	43
10.2.2 Time-of-flight	44
10.3 An y -dependent velocity field	45
10.4 Sensitivity to perturbation of gridcells	45
10.5 Time-of-flight	50
10.5.1 The single cell	50
10.5.2 Accumulation error in a skew grid	51
10.6 Parallelograms and the MPFA-method	52
11 Summary and Conclusions	57
11.1 Conclusion	57
11.2 Further work	58
A Streamline Simulation	59
A.1 The Governing IMPES Equations	59
A.2 Coordinate Transformation Along Streamlines	59
B Pollock tracing in 2D	61
B.1 Find the travel time and the entry edge	62
B.2 Find the entry point and the trajectory	67
B.3 Forward tracing	68
B.4 The special case of a source or a sink	68

C	Tracing on irregular grids	73
D	Determining if a point is inside a quadrilateral	85
E	The inverse bilinear transformation	91
	E.1 Case 1	92
	E.2 Case 2	93
	E.3 Case 3	93
	E.4 Case 4	93
	E.5 Case 5	94
F	The flux over an edge of a quadrilateral	95
G	The uniform velocity field, additional computations	97
	G.1 Flux computations	97
	G.2 Analytical tracing	99
	G.3 Analytical TOF	101
H	The circular velocity field, additional computations	102
	H.1 Computation of fluxes	102
I	The y-dependent velocity field	105
J	Calculation of time-of-flight	106
	J.1 The single cell	106
	J.2 The skew grid	107
K	Generation of a random convex grid.	109
L	Fluid particle	110
	L.1 Representative Elementary Volume	110
	L.2 Fluid particle and velocity	111
M	Area of quadrilateral in 2D	112
N	C++ Implementation	115
	N.1 Source	116
	N.1.1 Point inside quadrilateral	116
O	Matlab visualization	118
	O.1 Cell arrays	118
	O.2 Plotting streamlines	118

Chapter 1

Introduction

A streamline is a curve in space at a given time, whose tangent at a point is equal to the velocity at that point. Streamlines can thus be used to visualize the “directional” information in a velocity field at a given time.

The computation of streamlines is dependent on a precise description of the velocity field. When the velocity field is calculated numerically, it will be given at certain points or the flux of it is determined over certain faces, and interpolation is needed to give a complete field. The streamlines traced from such a velocity field will contain errors from both the numerical solution and from the interpolation method.

Within reservoir simulation, streamline tracing is an important element in the streamline method [5, 30, 15]. Streamline simulation has become increasingly popular in the recent years, because it allows for a fast evaluation of reservoir performance [30]. Accurate resolution of the streamlines permits large timesteps to be taken when solving the pressure equation.

An effective method for streamline tracing, developed by Pollock [22], involves the use of a piece-wise linear approximation of velocity with respect to each gridblock. The method gives as a by-product the time-of-flight [9]. The time-of-flight (TOF) is the time required to travel a given distance along a streamline based on the velocity field along the streamline, and it is used to define a coordinate system for the streamline method. It can also be used for reservoir performance prediction [15].

Pollock tracing, used in most 3D streamline simulators, was until recently restricted to grids with orthogonal Cartesian gridblocks. This has been a limitation of the method, since modeling of complex features such as faults or deviated wells generally require the use of structured or unstructured curvilinear grids. An extension of the method to such grids has been developed by, e.g., *Prévost et al.* [24]. In this thesis we will denote this method of tracing by the Prévost-method.

For a 2D grid, edgefluxes for each gridcell is transformed to a reference space consisting of unit square cells. Then fluxes are extended to the entire

cell by interpolation in the lowest order Raviart-Thomas space. Pollock tracing is then performed on a unit square cell in the reference space, and the streamline is at last mapped back to the physical space using a bilinear transformation.

In this thesis we will investigate the Prévost-method for the case of a 2D, logically, Cartesian grid in conjunction with the MPFA-method [1] for calculating fluxes or by imposing some known velocity fields on the grids. We will examine how sensitive the shape of the streamlines is with respect to perturbations of the grid in physical space.

The Prévost-method uses a flux mapping from physical space to the reference space that is based on a simple scaling of the fluxes by the Jacobi determinant of the midpoint of the cell. We will show that this method of flux mapping can give incorrect time-of-flight.

By introducing the Piola transformation, we also show that the method reproduces uniform flow. The Piola transformation is a vector transformation between the physical space and the reference space that preserves fluxes. Using a result in given in [19] for the Piola transformation we can show that the shape of the streamlines for a uniform flow field is exactly reproduced.

Additionally, we will see that streamlines can be used to evaluate the quality of a numerical velocity field. The velocity field will be computed using an MPFA-algorithm [1] for a simple flow problem. It is known that for very distorted grids, even the MPFA-method has difficulties calculating the correct fluxes. In this thesis we will investigate such effects using streamlines.

In Chapter 2 we give a brief introduction to the concepts of reservoir simulation. In Chapter 3 we introduce the flow equations for streamline simulation. Chapter 4 discusses the concept of a streamline. Chapter 5 defines the grids to be used in this work. In Chapter 6 we introduce the bilinear transformation and in Chapter 7 we define the time-of-flight. Chapter 8 describes the tracing algorithm, while Chapter 9 introduces the Piola transformation and we show that uniform flow streamlines are reproduced. Chapter 10 tests the tracing algorithm, and presents the results. In Chapter 11 we give a summary and conclusions based on the work presented.

In the appendices we have collected more detailed calculations and some topics are discussed in more depth.

Chapter 2

Basic concepts

2.1 Flow through porous media

Although streamline tracing can be done for any problem giving a velocity field, the usage in this thesis will be for flow in porous media.

Flow through porous media, [7], is a topic encountered in many branches of engineering and science, e.g., ground water hydrology, reservoir engineering, soil science, soil mechanics and chemical engineering.

In the most general sense, a porous material is a solid containing holes. The portion of the rock not occupied by solid matter is the void space (pore space). Only the connected pores are of interest for flow through porous media.

An example of a porous media is the aquifer. The aquifer is a geologic formation, that contains water, and permits significant amounts of water to move through it under ordinary field conditions. Ground water is a term used to denote all waters found beneath the ground surface.

An other example of porous media is the oil or gas reservoir. This is a porous geologic formation that contains in its pore space, in addition to water, at least one hydrocarbon (oil or gas) in a liquid or gaseous phase.

2.2 Reservoir simulation

Reservoir simulation is the process of inferring the behavior of a petroleum reservoir from the performance of a model [21], in order to optimize the recovery of hydrocarbon.

There are five major stages to the modelling process for computational science, and in particular to reservoir simulation:

- First, a physical model is developed
- Then a mathematical model of the physical model is obtained

- Next, a numerical scheme for solution of the mathematical model is formed
- Then a computer program is sought to implement the numerical model.
- And at last: Examination and validation of the first three steps based on the results from the computer model.

Chapter 3

Flow and transport equations

3.1 The IMPES Equations

Streamline tracing is an important component of the streamline method for reservoir simulation. A summary of the streamline method is given in Appendix A. Here we will derive the pressure and saturation equations for multiphase flow in the form used by the streamline method. In this thesis we will focus on one phase flow problems and the derivation will include such problems as a special case.

The streamline method is based on solving first for the pressure field and then for the saturation distribution [6]. This is an IMplicit in Pressure, EXplicit in Saturation method (IMPES). One advantage of a conventional IMPES formulation over the fully-implicit formulation is that numerical diffusion due to discretization error is reduced. The trade-off is that smaller time step sizes must be taken due to stability considerations.

The governing equation for flow of a component i with n_p phases flowing in porous medium is defined by *Lake* [16] as,

$$\sum_{j=1}^{n_p} \left\{ \frac{\partial}{\partial t} (\phi \omega_{ij} \rho_j S_j) + \nabla \cdot (\omega_{ij} \rho_j \mathbf{u}_j - \phi \rho_j S_j \tilde{\mathbf{D}}_{ij} \cdot \nabla \omega_{ij}) \right\} = \sum_{j=1}^{n_p} q_s \rho_j \omega_{ij} \quad (3.1)$$

where q_s represents a source or sink volume flow rate, $\tilde{\mathbf{D}}_{ij}$ characterizes the component dispersivity, ω_{ij} is the mass fraction of component i in phase j , and \mathbf{u}_j is the phase velocity given by Darcy's law,

$$\mathbf{u}_j = -\frac{\tilde{\mathbf{K}} k_{rj}}{\mu_j} \cdot (\nabla P_j - \rho_j \mathbf{g}) \quad (3.2)$$

The phase pressure is P_j , \mathbf{g} is the gravitational vector, which is assumed constant and directed toward the earth's center. To simplify (3.1) it is assumed that the fluids are incompressible ($\rho_j = \text{constant}$) and there is no

dispersivity ($\tilde{\mathbf{D}}_{ij} = 0$) giving,

$$\sum_{j=1}^{n_p} \left\{ \frac{\partial}{\partial t} (\phi \omega_{ij} S_j) + \nabla \cdot \omega_{ij} \mathbf{u}_j \right\} = \sum_{j=1}^{n_p} q_s \omega_{ij} \quad (3.3)$$

Next, summing Equation (3.3) over all the components and using the fact that $\sum_{i=1}^{n_p} \omega_{ij} = 1$ gives,

$$\nabla \cdot \mathbf{u}_t = q_s \quad (3.4)$$

the governing volume balance equation for incompressible flow. The total velocity \mathbf{u}_t can be defined by summing Equation (3.2) over n_p phases to give,

$$\mathbf{u}_t = -\tilde{\mathbf{K}} \cdot (\lambda_t \nabla P - \lambda_g \mathbf{g}) \quad (3.5)$$

Capillary pressure has been neglected such that $P = P_j$. The total mobility (λ_t) and the total gravity mobility are defined as,

$$\lambda_t = \sum_{j=1}^{n_p} \frac{k_{rj}}{\mu_j}, \quad \lambda_g = \sum_{j=1}^{n_p} \frac{k_{rj} \rho_j}{\mu_j} \quad (3.6)$$

Finally, combining Equation (3.4) and Equation (3.5) leads to the governing pressure equation for multiphase incompressible flow in porous media,

$$\nabla \cdot \tilde{\mathbf{K}} \cdot (\lambda_t \nabla P - \lambda_g \mathbf{g}) = -q_s \quad (3.7)$$

Equation (3.7) is elliptic and is known as a Poisson equation where the unknown is P . It is coupled to the saturation equation, (3.13), through λ_t and λ_g . Once P is defined, \mathbf{u}_t can be determined by using Equation (3.5).

The governing saturation equation for the IMPES method can be derived from Equation (3.3). To simplify the problem, it is assumed that the phases are immiscible such that $\omega_{ij} = 0$ for $i \neq j$ and $\omega_{ij} = 1$ for $i = j$ giving,

$$\phi \frac{\partial S_j}{\partial t} + \nabla \cdot \mathbf{u}_j = q_s f_{j,s} \quad (3.8)$$

where ϕ is assumed constant and $f_{j,s}$ is the fraction contributed by phase j to q_s . Substituting Darcy's Law, Equation (3.2), into Equation (3.8) and eliminating ∇P by using Equation (3.5) the above equation becomes,

$$\phi \frac{\partial S_j}{\partial t} + \nabla \cdot \left(\frac{k_{rj}/\mu_j}{\lambda_t} \mathbf{u}_t - \tilde{\mathbf{K}} \cdot \mathbf{g} \frac{k_{rj}/\mu_j}{\lambda_t} \sum_{m=1}^{n_p} k_{rm}/\mu_m (\rho_m - \rho_j) \right) = q_s f_{j,s} \quad (3.9)$$

Defining the standard Buckley-Leverett fractional flow term as,

$$f_j = \frac{k_{rj}/\mu_j}{\lambda_t} \quad (3.10)$$

and a gravity fractional flow term given by,

$$\tilde{\mathbf{G}}_j = -\tilde{\mathbf{K}} \cdot \mathbf{g} \frac{k_{rj}/\mu_j}{\lambda_t} \sum_{m=1}^{n_p} k_{rm}/\mu_m (\rho_m - \rho_j) \quad (3.11)$$

Equation (3.8) can be rewritten as

$$\phi \frac{\partial S_j}{\partial t} + \nabla \cdot f_j \mathbf{u}_t + \nabla \cdot \tilde{\mathbf{G}}_j = q_s f_{j,s} \quad (3.12)$$

Given that $\nabla \cdot \mathbf{u}_t = 0$ for incompressible flow, the governing saturation Equation of an individual phase becomes

$$\phi \frac{\partial S_j}{\partial t} + \mathbf{u}_t \cdot \nabla f_j + \nabla \cdot \tilde{\mathbf{G}}_j = q_s f_{j,s} \quad (3.13)$$

The saturation equation is hyperbolic.

3.2 Numerical solution: MPFA

For some of the testcases we will study single-phase incompressible horizontal flow in 2D. Then the continuity Equation (3.3) reduces to

$$\nabla \cdot \mathbf{u} = q_s \quad (3.14)$$

where \mathbf{u} is the volumetric flux density given by Darcy's law (3.2),

$$\mathbf{u} = -\frac{\tilde{\mathbf{K}}}{\mu} \nabla P \quad (3.15)$$

Taking the divergence of (3.15) and substituting (3.14) gives

$$\nabla \cdot \left(\frac{\tilde{\mathbf{K}}}{\mu} \nabla P \right) = -q_s \quad (3.16)$$

Since all the variables in (3.16) are assumed to be invariant with time, we will have steady-state flow, and streamlines are constant in time.

Techniques for analytical solution of such problems exists [14, 18], and this can be used to check both the numerical solution of (3.16) as well as the streamline tracing algorithm.

For the numerical solution of (3.16) we will use a MPFA-method as described in [1]. We discretize the solution domain using an irregular quadrilateral grid. The method then constructs a linear system

$$\tilde{\mathbf{A}} \mathbf{u} = \mathbf{q} \quad (3.17)$$

where \mathbf{u} is a vector of the unknown cellcentre pressures and \mathbf{q} is a vector contain source terms. The system can be solved by standard methods to give the cellcentre pressures. Then the fluxes over celledges can be calculated from Darcy's law. These fluxes are then used to compute streamlines.

Chapter 4

Streamlines

The central point of this chapter is to define the *pathline* and the *streamline*. We will also give an introduction to the stream function in two dimensions. Although we will not use the stream function in this thesis, it shows that for certain ideal cases the streamlines might be found analytically.

4.1 Eulerian and Lagrangian representations of fluid motion

Fluid motion is determined completely if its *velocity* \mathbf{v} , pressure p , density ρ , temperature and other possible parameters are specified as functions of coordinates and time. This is the Eulerian way of representing fluid motion, where we fix our attention to a point in space and follow the time variation of \mathbf{v} , p , etc., at this point. Alternatively, if we fix time we can determine the spatial variation of the same parameters. However there is no direct information about the motion of a given *fluid particle* in this representation.

In a Lagrangian representation, on the other hand, all parameters including the coordinates of the fluid's particles $x_i (i = 1, 2, 3)$ are specified in terms of time t and some vector (ξ_1, ξ_2, ξ_3) which identifies a given particle of the fluid: $x_i = x_i(\xi_k, t)$, $p = p(\xi_k, t)$, $\rho = \rho(\xi_k, t)$, etc. The variables (ξ_1, ξ_2, ξ_3) are usually assumed to be the coordinates of a fluid particle at some time t_0 (initial coordinates), i.e. $\xi_i = x_i(\xi_k, t_0)$. Therefore, in a Lagrangian representation, our attention is fixed on a chosen fluid particle.

A more thorough description of fluid particle and velocity is given in Appendix L.

4.2 Pathlines and Streamlines

Instead of observing pressure variations in an experiment and relating the flux to them, one may start by observing streamlines in an experiment of

steady flow through a porous medium domain, and then relate the flux to these observations, [7]. The streamline, as well as the pathline, is to be understood as an average concept.

In practical terms, it is impossible to label a single fluid particle (say in an experiment of flow through a porous medium) and observe its motion. Instead we label a group of particles occupying a small neighbourhood, or we continuously inject a tracer into a point in a steadily moving fluid. In laminar flow, in spite of hydrodynamic dispersion, and in the case of continuous injection, in spite of the lateral dispersion, it is possible to define the average path of the particles and use it in describing the flow [7].

A *pathline* of a fluid particle is the locus of its position in space as time passes. It is thus the trajectory of a particle of fixed identity. The pathline is described by the solutions of three parametric equations:

$$\frac{dx}{V_x(x, y, z, t)} = \frac{dy}{V_y(x, y, z, t)} = \frac{dz}{V_z(x, y, z, t)} = dt \quad (4.1)$$

In flow through porous media, $\mathbf{V} = (V_x, V_y, V_z)$ are the average velocity vector, with $\mathbf{V} = \frac{1}{\phi}\mathbf{u}$, where \mathbf{u} is the Darcy velocity, defined in (3.2), for a given phase.

At any instant of time, there is at every point in a flow domain a velocity vector, \mathbf{q} , with a definite direction. The instantaneous curves that are at every point tangent to the direction of the velocity at that point are called *streamlines* of the flow. The mathematical expression defining a streamline is therefore $\mathbf{q} \times d\mathbf{r} = 0$, or:

$$\frac{dx}{q_x(x, y, z, t_0)} = \frac{dy}{q_y(x, y, z, t_0)} = \frac{dz}{q_z(x, y, z, t_0)} \quad (4.2)$$

Here t_0 indicates a certain time, and $d\mathbf{r}$ is an element of arc along a streamline. For flow through porous media the streamlines are the same if use $\mathbf{q} = \mathbf{V}$ or $\mathbf{q} = \mathbf{u}$ (see Section 8.3). Equation (4.2) is valid for both isotropic and anisotropic media. Consider flow described by (3.15)

$$\mathbf{u} = \frac{\tilde{\mathbf{K}}}{\mu} \nabla P \quad (4.3)$$

If x, y, z are the principal directions of permeability, equation (4.2) then becomes:

$$\frac{\mu dx}{K_x \partial P / \partial x} = \frac{\mu dy}{K_y \partial P / \partial y} = \frac{\mu dz}{K_z \partial P / \partial z} \quad (4.4)$$

For an isotropic medium, $K_x = K_y = K_z = K$ in (4.4).

Had we started from postulating the presence of an equipotential surface $P = \text{const}$, we would consider an elementary displacement $d\mathbf{s}$ in the direction normal to this surface. Then:

$$\nabla P \times d\mathbf{s} = 0; \quad \frac{dx}{\partial P / \partial x} = \frac{dy}{\partial P / \partial y} = \frac{dz}{\partial P / \partial z} \quad (4.5)$$

which define curves in space normal to the equipotential surfaces. These are the streamlines. Multiplying (4.4), written for an isotropic medium, by K/μ , we obtain (4.5). Thus, in an isotropic medium, streamlines are perpendicular to the equipotential surfaces. These two independent approaches to the definition of streamlines are possible; both are based on observations of field phenomena.

In steady flow, i.e., one in which flow characteristics remain invariant with time, streamlines and pathlines coincide. In unsteady flow they may be distinct. In unsteady flow we can speak only of an instantaneous picture of the streamlines, as the picture varies continuously.

Equation (4.2) can be written

$$q_y dx - q_x dy = 0 \quad (4.6a)$$

$$q_x dz - q_z dx = 0 \quad (4.6b)$$

$$q_z dy - q_y dz = 0 \quad (4.6c)$$

Let $\mathbf{q} \neq 0$ and assume $q_x \neq 0$, then regarding y and z as functions of x , we get

$$\frac{dy}{dx} = \frac{q_y}{q_x} \quad (4.7a)$$

$$\frac{dz}{dx} = \frac{q_z}{q_x} \quad (4.7b)$$

This is a system of ordinary differential equations where t is a parameter. We can find the streamline passing through (x_0, y_0, z_0) at t , by solving (4.7) for the given starting condition. Then if $\mathbf{q} \neq 0$ there can only pass one streamline through a point at a given time. If $\mathbf{q} = 0$ we have a stationary point, through which there may pass several streamlines [10].

4.3 The stream function

Assume that we have two-dimensional flow of an incompressible fluid, and that the motion is the same in all planes parallel to the xy -plane.

Then a stream function may be derived by assuming the existence of pathlines and streamlines. The flux may be expressed in terms of this stream function [7].

Assuming no sources, the continuity equation, (3.3) becomes

$$\nabla \cdot \mathbf{q} = 0 \quad (4.8)$$

Which is satisfied if we write

$$\mathbf{q} = \nabla \times \psi(x, y, t)\mathbf{k} = \frac{\partial \psi}{\partial y}\mathbf{i} - \frac{\partial \psi}{\partial x}\mathbf{j} \quad (4.9)$$

where $\psi(x, y, t)$ is the stream function. The streamlines for this case is given by (4.6a),

$$q_x dy - q_y dx = 0 \quad (4.10)$$

From (4.9) we have that $q_x = \frac{\partial \psi}{\partial y}$ and $q_y = -\frac{\partial \psi}{\partial x}$. Using this in (4.10), we get

$$\frac{\partial \psi}{\partial x} dx + \frac{\partial \psi}{\partial y} dy = d\psi = 0 \quad (4.11)$$

This can be integrated to give

$$\psi(x, y, t) = \text{constant} \quad (4.12)$$

along a streamline.

Now assume in addition irrotational flow,

$$\nabla \times \mathbf{q} = 0 \quad (4.13)$$

Taking the curl of (4.9) and using (4.13), we find that

$$\nabla^2 \psi = 0 \quad (4.14)$$

For certain boundary conditions this equation can be solved analytically.

Equation (4.13) implies that the velocity can be written as the gradient of a potential,

$$\mathbf{q} = \nabla \phi = \frac{\partial \phi}{\partial x} \mathbf{i} + \frac{\partial \phi}{\partial y} \mathbf{j} \quad (4.15)$$

Then comparing the two expressions for the velocity (4.9) and (4.15), we get

$$\frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \quad (4.16)$$

$$\frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x} \quad (4.17)$$

This is the Cauchy-Riemann equations which implies

$$\nabla \phi \cdot \nabla \psi = 0 \quad (4.18)$$

so streamlines and equipotential lines are orthogonal in two-dimensional incompressible, irrotational flow.

Chapter 5

Irregular Grids

A reservoir simulator predicts reservoir performance by solving flow equations on a discrete grid chosen to represent the reservoir. Flow simulations of reservoirs with complex geometries and geological features generally require the use of flexible structured and unstructured grids in order to resolve important features such as faults and deviated wells [4, 31]. Corner-point geometry [23] (or nonorthogonal grids) are widely used in reservoir simulation.

These grids are structured (logically Cartesian) and are comprised of quadrilateral cells. Unstructured grids are generally comprised of triangular and/or quadrilateral cells (in 2D) and are generally based on a more complex data structure than the standard logically Cartesian grid indices.

5.1 Structured logically Cartesian grids

We will use a structured logically Cartesian grid for the tests in this thesis. Note that the Prévost-method has been extended to unstructured grids [24]. Before we can define the grid structure we have to make some definitions.

Consider a quadrilateral Q . See Figure 5.1. We denote the corners of Q by \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 and \mathbf{x}_4 . Q is not uniquely determined by its four corners alone. See Figure 5.2. We have to specify an ordering of the points to get a unique quadrilateral.

Definition 5.1 (Quadrilateral) *With a quadrilateral $Q = Q(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ we will mean the quadrilateral with corners $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, and \mathbf{x}_4 such that if starting at the corner \mathbf{x}_1 and walking around the border of Q we will meet the other corners in the sequence $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, and \mathbf{x}_4 or in the sequence $\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_3$, and \mathbf{x}_2 depending on the direction chosen.*

Definition 5.2 (Convex quadrilateral) *In a convex quadrilateral all internal angles are less than 180° .*

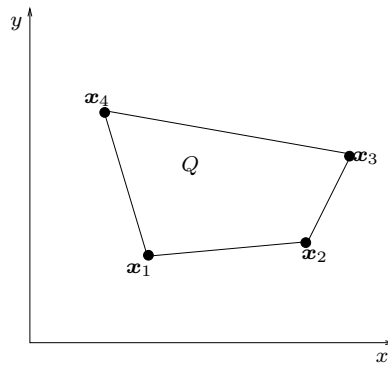
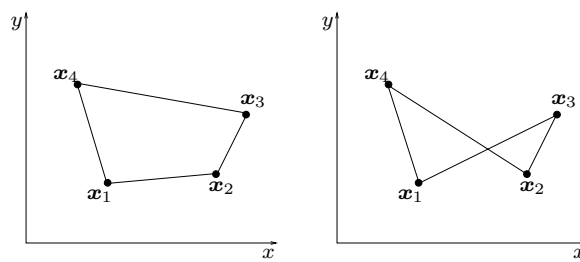


Figure 5.1: A quadrilateral.

Figure 5.2: There are two ways of interpreting the four points x_1 , x_2 , x_3 , x_4 .

Definition 5.3 (Irregular convex quadrilateral) *By an irregular convex quadrilateral, Q , we shall mean a convex quadrilateral that is not a square or a rectangle.*

Definition 5.4 (Edges of a quadrilateral) *The boundary of $Q(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ consists of four edges. Let E_{ij} be the edge that contains \mathbf{x}_i and \mathbf{x}_j . We shall also use $A = E_1 = E_{14}$, $B = E_2 = E_{23}$, $C = E_3 = E_{12}$ and $D = E_4 = E_{34}$.*

Walking around the boundary of Q , starting at an A -edge we should encounter the edges of Q in the sequence A, C, B, D or A, D, B, C .

The grid consists of $p_x p_y$ gridpoints numbered from 0 to $p_x p_y - 1$, such that for a given gridpoint number p that satisfies

$$p < p_x(p_y - 1) - 1 \quad \wedge \quad (p + 1) \bmod p_x \neq 0 \quad (5.1)$$

the gridpoints numbered $p, p + 1, p + p_x + 1$ and $p + p_x$ makes a convex quadrilateral $Q(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$. Where \mathbf{x}_1 corresponds to gridpoint p , \mathbf{x}_2 to $p + 1$, \mathbf{x}_3 to $p + p_x + 1$ and \mathbf{x}_4 to $p + p_x$. See Figure 5.3 for an example.

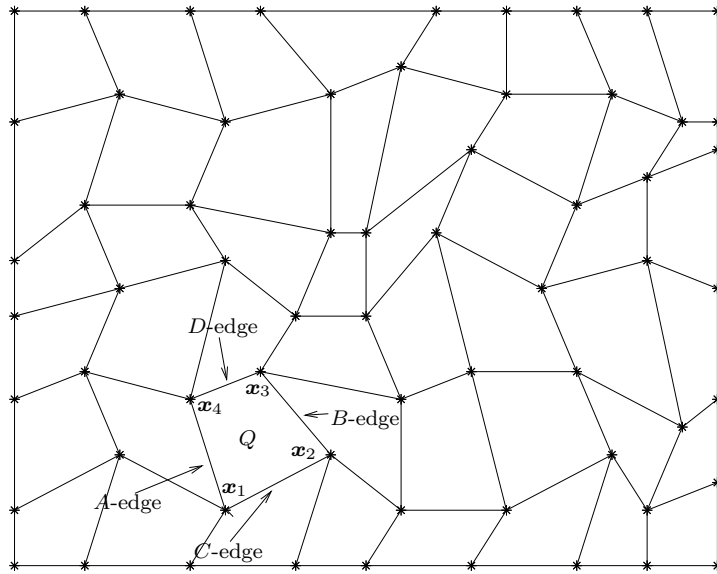


Figure 5.3: Sample grid with gridpoints marked with '*'

With Q there is associated a gridcell number, c . Let p be the lowest gridpoint number in Q , then

$$c = (p_x - 1)\text{floor}(p/p_x) + p \bmod p_x \quad (5.2)$$

where $\text{floor}(x)$ rounds x down to the nearest integer. See Figure 5.4.

Another example is presented in Figure 5.5. We see that we also must require that the gridpoints are such that no quadrilaterals overlap another. The grid in Figure 5.5 is allowed. But note that there is no connection over the A -edge of gridcell 0 or the B -edge of gridcell 10.

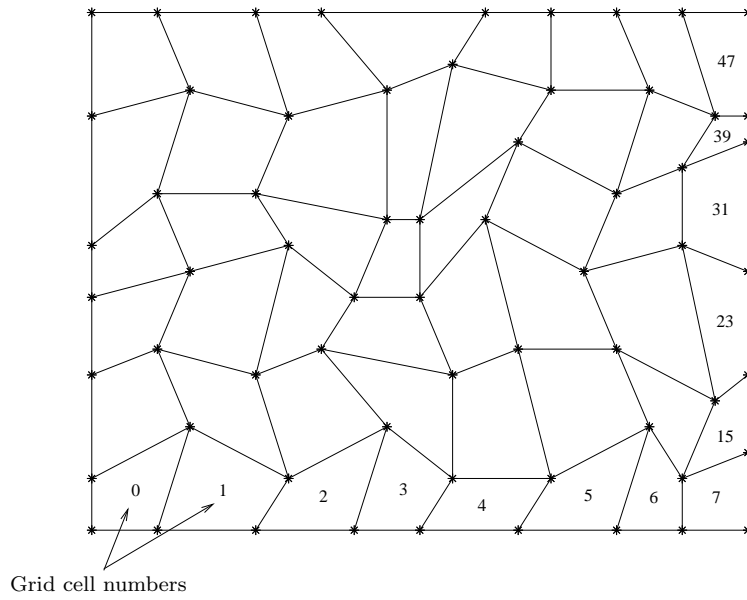


Figure 5.4: Sample grid with gridcell numbers

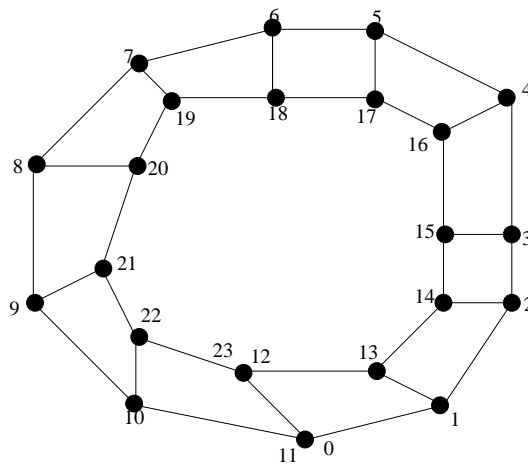


Figure 5.5: An unusual grid.

Chapter 6

The Bilinear Transformation

6.1 The transformation

An important component of the streamline methods to be described in the following chapters is the isoparametric mapping [8] of the streamlines from a reference space to the physical space. In 2D the mapping can be viewed as a bilinear mapping of quadrilaterals. See Figure 6.1.

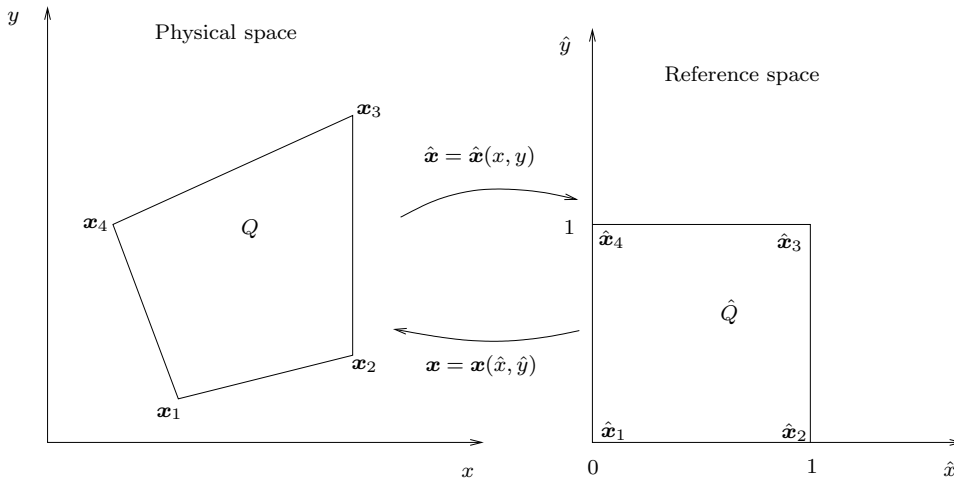


Figure 6.1: Quadrilaterals are mapped to a unit square through a bilinear transformation.

We denote the the simulation grid containing the irregular quadrilaterals by the physical space, \mathcal{P} , and the reference space by \mathcal{R} . The quadrilateral in \mathcal{P} corresponding to \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 and \mathbf{x}_4 will be referred to as $Q = Q(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ (see definition 5.1), and the corresponding unit square in \mathcal{R} will be \hat{Q} . Variables in \mathcal{R} will usually be denoted by a '^', e.g. \hat{Q} .

The transformation from \mathcal{R} to \mathcal{P} is given by

$$\begin{aligned} \mathbf{x}(\hat{\mathbf{x}}) = \begin{bmatrix} x(\hat{x}, \hat{y}) \\ y(\hat{x}, \hat{y}) \end{bmatrix} = \\ \mathbf{x}_1(1 - \hat{x})(1 - \hat{y}) + \mathbf{x}_2\hat{x}(1 - \hat{y}) + \mathbf{x}_3\hat{x}\hat{y} + \mathbf{x}_4(1 - \hat{x})\hat{y} = \\ \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} (1 - \hat{x})(1 - \hat{y}) + \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \hat{x}(1 - \hat{y}) + \\ \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \hat{x}\hat{y} + \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} (1 - \hat{x})\hat{y} \quad (6.1) \end{aligned}$$

where (\hat{x}, \hat{y}) are the local reference space coordinates such that $0 \leq \hat{x} \leq 1$ and $0 \leq \hat{y} \leq 1$.

6.2 Properties

6.2.1 Mapping of straight lines

Curves in \mathcal{R} corresponding to a constant value of \hat{x} or \hat{y} , \hat{x} -curves or \hat{y} -curves respectively, will be mapped to straight lines in \mathcal{P} by the transformation. See Figure 6.2. To see this, note that the transformation (6.1) for a constant value of \hat{y} can be written

$$\mathbf{x}(\hat{\mathbf{x}}) = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a\hat{x} + b \\ c\hat{x} + d \end{bmatrix} \quad (6.2)$$

for some constants a, b, c, d . From this we get that $\hat{x} = (x - b)/a$, and then $y = c(x - b)/a + d$ which is a straight line in \mathcal{P} . The same argument holds for a constant value of \hat{y} .

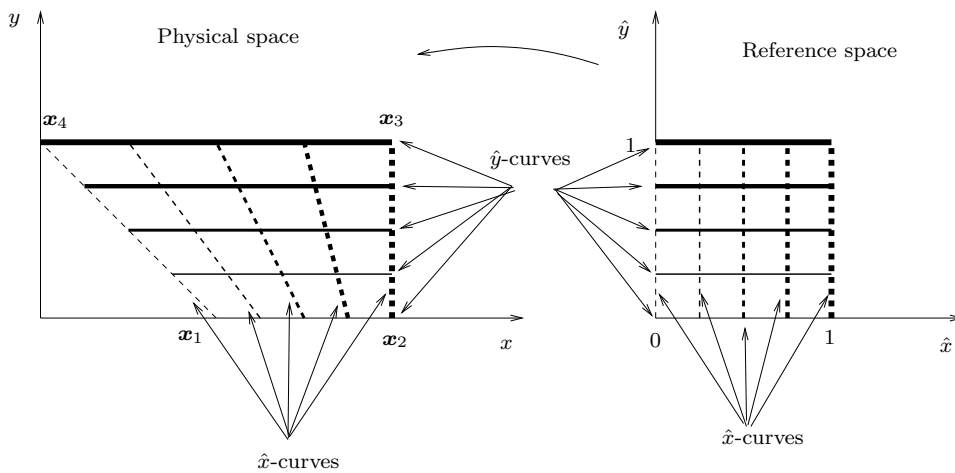


Figure 6.2: Mapping of \hat{x} -curves and \hat{y} -curves.

For \mathcal{R} we define $\hat{\mathbf{x}}_1 = (0, 0)$, $\hat{\mathbf{x}}_2 = (1, 0)$, $\hat{\mathbf{x}}_3 = (1, 1)$ and $\hat{\mathbf{x}}_4 = (0, 1)$, and,

$$\hat{A}\text{-edge: } \{\text{The edge containing } \hat{\mathbf{x}}_4 \text{ and } \hat{\mathbf{x}}_1.\} \quad (6.3)$$

$$\hat{C}\text{-edge: } \{\text{The edge containing } \hat{\mathbf{x}}_1 \text{ and } \hat{\mathbf{x}}_2.\} \quad (6.4)$$

$$\hat{B}\text{-edge: } \{\text{The edge containing } \hat{\mathbf{x}}_2 \text{ and } \hat{\mathbf{x}}_3.\} \quad (6.5)$$

$$\hat{D}\text{-edge: } \{\text{The edge containing } \hat{\mathbf{x}}_3 \text{ and } \hat{\mathbf{x}}_4.\} \quad (6.6)$$

It is easy to see that $\hat{\mathbf{x}}_i$ gets mapped to \mathbf{x}_i , $i = 1, 2, 3, 4$. Then since straight lines get mapped to straight lines and by Definition 5.4, we must have that an i -edge gets mapped to an \hat{i} -edge, $i = A, B, C, D$.

6.2.2 The midpoint

We refer to $\hat{\mathbf{x}}_m = (\hat{x}_m, \hat{y}_m) = (1/2, 1/2)$ as the midpoint of \hat{Q} . If we substitute $\hat{\mathbf{x}}_m$ in (6.1) we get

$$x_m \equiv x = \frac{1}{4}(x_1 + x_2 + x_3 + x_4), \quad y_m \equiv y = \frac{1}{4}(y_1 + y_2 + y_3 + y_4) \quad (6.7)$$

We will refer to $\mathbf{x}_m = (x_m, y_m)$ as the *midpoint* of Q .

6.2.3 The Jacobi matrix and the Jacobi determinant

The Jacobi matrix of the transformation, (6.1), is given by

$$\tilde{D}(\hat{\mathbf{x}}) = \frac{d\mathbf{x}}{d\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{bmatrix} = \begin{bmatrix} (x_2 - x_1)(1 - \hat{y}) + (x_3 - x_4)\hat{y} & (x_4 - x_1)(1 - \hat{x}) + (x_3 - x_2)\hat{x} \\ (y_2 - y_1)(1 - \hat{y}) + (y_3 - y_4)\hat{y} & (y_4 - y_1)(1 - \hat{x}) + (y_3 - y_2)\hat{x} \end{bmatrix} \quad (6.8)$$

Then the Jacobi determinant is given by

$$\begin{aligned} J(\hat{\mathbf{x}}) = \det \tilde{D}(\hat{\mathbf{x}}) &= \frac{\partial x}{\partial \hat{x}} \frac{\partial y}{\partial \hat{y}} - \frac{\partial x}{\partial \hat{y}} \frac{\partial y}{\partial \hat{x}} = \\ &= (y_1x_3 - x_1y_3 + x_2y_3 - y_2x_3 + x_1y_4 - y_1x_4 + y_2x_4 - x_2y_4)\hat{x} \\ &+ (y_1x_2 - x_1y_2 + x_1y_3 - y_1x_3 + y_2x_4 - x_2y_4 + x_3y_4 - y_3x_4)\hat{y} \\ &+ x_1y_2 - y_1x_2 + x_2y_4 - y_2x_4 + y_1x_4 - x_1y_4 \quad (6.9) \end{aligned}$$

Observe that

$$J_m \equiv J(\hat{\mathbf{x}}_m) = \frac{1}{2}(x_1y_2 - y_1x_2 + x_2y_3 - y_2x_3 + y_1x_4 - x_1y_4 + x_3y_4 - y_3x_4) \quad (6.10)$$

Comparing this with Equation (M.20) we see that we must have that

$$|J_m| = \{\text{Area of } Q\} \quad (6.11)$$

The inverse of the Jacobimatrix is given by

$$\tilde{D}^{-1} = \frac{d\hat{\mathbf{x}}}{d\mathbf{x}} = \frac{1}{J} \begin{bmatrix} \frac{\partial y}{\partial \hat{y}} & -\frac{\partial x}{\partial \hat{y}} \\ -\frac{\partial y}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{x}} \end{bmatrix} \quad (6.12)$$

provided $J \neq 0$.

6.3 The inverse transformation

It is possible to calculate the inverse transformation, $\hat{\mathbf{x}} = \hat{\mathbf{x}}(\mathbf{x})$, (see Appendix E), but since the four parameters of the transformation, namely the points \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 and \mathbf{x}_4 may represent a single point, a line, a triangle, two triangles, a non-convex quadrialteral, the inverse is not straight forward. If the four points represent a convex quadrialteral, the inverse will always exist, since a sufficient and necessary condition is that the Jacobi determinant, J , satisfies $J \neq 0$ [2]. Since J is the ratio between two infinitesimal area elements in \mathcal{R} and \mathcal{P} under the bilinear transformation, this ratio cannot be zero for a convex Q .

6.4 Flux calculation

In this section we will derive an expression for the flux across an \hat{x} -curve or an \hat{y} -curve in \mathcal{P} . This will be used later to show that the Piola transformation conserves fluxes.

First we write the Jacobian, (6.1), as

$$\tilde{D}(\hat{\mathbf{x}}) = \frac{d\mathbf{x}}{d\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \hat{x}} & \frac{\partial \mathbf{x}}{\partial \hat{y}} \end{bmatrix} \quad (6.13)$$

and the inverse as

$$\tilde{D}^{-1} = \frac{1}{J} \begin{bmatrix} \frac{\partial y}{\partial \hat{y}} & -\frac{\partial x}{\partial \hat{y}} \\ -\frac{\partial y}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{x}} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} \boldsymbol{\nu}_1^T \\ \boldsymbol{\nu}_2^T \end{bmatrix} \quad (6.14)$$

Since the vectors $\frac{\partial \mathbf{x}}{\partial \hat{x}}$, $\frac{\partial \mathbf{x}}{\partial \hat{y}}$ are tangent vectors to the \hat{y} -curves and \hat{x} -curves, respectively, it follows that the vectors $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are normal vectors to the same curves since

$$\boldsymbol{\nu}_1 \cdot \frac{\partial \mathbf{x}}{\partial \hat{y}} = 0, \quad \boldsymbol{\nu}_2 \cdot \frac{\partial \mathbf{x}}{\partial \hat{x}} = 0 \quad (6.15)$$

see Figure 6.3.

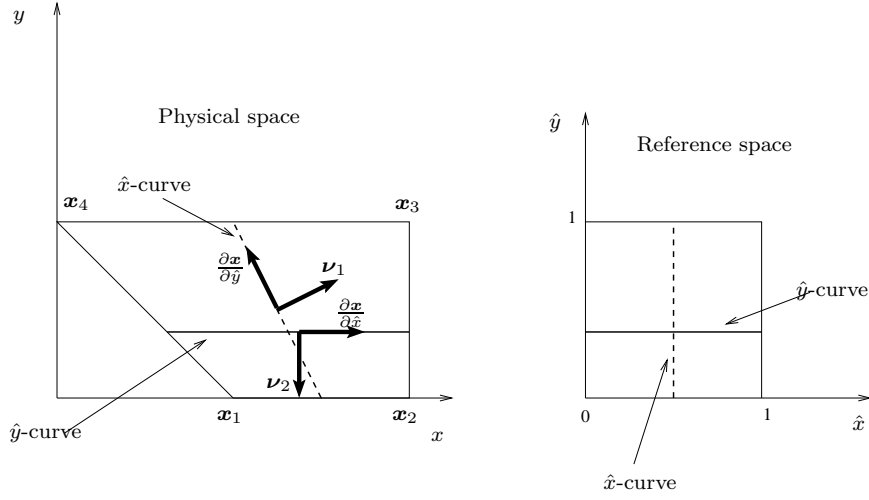


Figure 6.3: The bilinear transformation

From (6.14) and (6.13) we have that

$$\|\boldsymbol{\nu}_1\| = \sqrt{\left(\frac{\partial y}{\partial \hat{y}}\right)^2 + \left(-\frac{\partial x}{\partial \hat{y}}\right)^2} = \sqrt{\left(\frac{\partial x}{\partial \hat{y}}\right)^2 + \left(\frac{\partial y}{\partial \hat{y}}\right)^2} = \left\|\frac{\partial \mathbf{x}}{\partial \hat{y}}\right\| \quad (6.16)$$

$$\|\boldsymbol{\nu}_2\| = \sqrt{\left(-\frac{\partial y}{\partial \hat{x}}\right)^2 + \left(\frac{\partial x}{\partial \hat{x}}\right)^2} = \sqrt{\left(\frac{\partial x}{\partial \hat{x}}\right)^2 + \left(\frac{\partial y}{\partial \hat{x}}\right)^2} = \left\|\frac{\partial \mathbf{x}}{\partial \hat{x}}\right\| \quad (6.17)$$

Now the flux of a vector field \mathbf{q} across a curve $C(\hat{C}) = \mathbf{x}(\hat{\mathbf{r}}(\theta))$ in \mathcal{P} that is the image of a curve $\hat{C} = \hat{\mathbf{r}}(\theta)$ in \mathcal{R} is given by

$$\int_{C(\hat{C})} (\mathbf{q} \cdot \mathbf{N})(\mathbf{x}) ds = \int_a^b (\mathbf{q} \cdot \mathbf{N})(\mathbf{x}(\hat{\mathbf{r}}(\theta))) \left\|\frac{d\mathbf{x}}{d\theta}\right\| d\theta \quad (6.18)$$

where \mathbf{N} is the unit normal to the curve C , and, by the chain rule,

$$\left\|\frac{d\mathbf{x}}{d\theta}\right\| = \left\|\frac{d\mathbf{x}}{d\hat{\mathbf{r}}} \frac{d\hat{\mathbf{r}}}{d\theta}\right\| \quad (6.19)$$

Now consider a \hat{x} -curve in reference space given by

$$\hat{\mathbf{r}}(\theta) = \hat{a}\hat{\mathbf{i}} + \theta\hat{\mathbf{j}}, \quad 0 \leq \theta \leq 1 \quad (6.20)$$

where \hat{a} is a constant such that $0 \leq \hat{a} \leq 1$, then

$$\left\|\frac{d\mathbf{x}}{d\theta}\right\| = \left\|\frac{d\mathbf{x}}{d\hat{\mathbf{r}}} \frac{d\hat{\mathbf{r}}}{d\theta}\right\| = \left\|\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \hat{x}} & \frac{\partial \mathbf{x}}{\partial \hat{y}} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\| = \left\|\frac{\partial \mathbf{x}}{\partial \hat{y}}\right\| \quad (6.21)$$

From (6.15) and (6.16) it follows that

$$\mathbf{N} = \frac{\boldsymbol{\nu}_1}{\|\boldsymbol{\nu}_1\|} = \frac{\boldsymbol{\nu}_1}{\left\|\frac{\partial \mathbf{x}}{\partial \hat{y}}\right\|} \quad (6.22)$$

So by (6.18), the flux, $f_{\hat{x}}$, over a \hat{x} -curve then becomes

$$f_{\hat{x}} = \int_0^1 (\mathbf{q} \cdot \frac{\boldsymbol{\nu}_1}{\|\frac{\partial \mathbf{x}}{\partial \hat{y}}\|})(\mathbf{x}(\hat{\mathbf{r}}(\theta))) \|\frac{\partial \mathbf{x}}{\partial \hat{y}}\| d\theta = \int_0^1 (\mathbf{q} \cdot \boldsymbol{\nu}_1)(\mathbf{x}(\hat{\mathbf{r}}(\theta))) d\theta \quad (6.23)$$

A similar computation shows that the flux, $f_{\hat{y}}$, over a \hat{y} -curve is given by

$$f_{\hat{y}} = \int_0^1 (\mathbf{q} \cdot \boldsymbol{\nu}_2)(\mathbf{x}(\hat{\mathbf{s}}(\theta))) d\theta \quad (6.24)$$

where $\hat{\mathbf{s}}(\theta)$ is a representation for an \hat{y} -curve.

Chapter 7

The Time-of-Flight

7.1 Time-of-Flight

The time-of-flight (TOF) [15] is the time required to reach a distance, s , along a streamline based on the velocity field along the streamline. For the streamline method (Appendix A) it is defined as

$$\tau(s) = \int_0^s \frac{\phi(\zeta)}{|\mathbf{u}_t(\zeta)|} d\zeta \quad (7.1)$$

where \mathbf{u}_t is the total Darcy velocity given in (3.5) and ϕ is the porosity.

For the testcases we will assume 2D, one phase flow, and either use $\phi(x, y) = 1$ and $\mathbf{u} = \mathbf{u}_t$, or we consider velocity fields, \mathbf{u} , that are not associated with a flow problem in porous medium. We then define the TOF along a streamline, or piece of a streamline, C , as the line integral

$$\tau = \int_C \frac{1}{|\mathbf{u}(x, y, t_0)|} ds \quad (7.2)$$

The above integrals can be evaluated analytically using Pollock tracing described in Section 8.2.

Chapter 8

Streamline Tracing

In this chapter, we first briefly review different tracing methods for orthogonal Cartesian gridblocks. Then Pollock tracing is described in detail. Next, we consider tracing on irregular grids and how to define a consistent tracing velocity in \mathcal{R} . At last we sum up the Prévost-method for tracing on irregular grids.

8.1 Streamline Tracing

As shown in Section 4.2, in steady state flow tracing streamlines is equivalent to particle tracking. The majority of published work on streamlines has been in the groundwater literature [22, 27, 11, 17, 29, 26]. The methods use particle tracking ideas to define a streamline.

The particle's position for horizontal flow is given by (4.1) as

$$\int \frac{dx}{V_x} = \int dt \quad (8.1a)$$

$$\int \frac{dy}{V_y} = \int dt \quad (8.1b)$$

where $V_x = V_x(x, y)$ and $V_y = V_y(x, y)$ for steady flow.

The first step in a particle tracking procedure is to solve the flow equations, which gives velocities at cellcenters of the simulation grid.

Now, the integration of (8.1) can be accomplished by one of three approaches: analytical, numerical, or semianalytical. The analytical approach produces exact solutions, but only for a limited number of ideal cases of steady flow, homogeneous media, and simple geometry. In practice, analytical solutions are mainly used to validate approximate methods.

Numerical techniques for integration include the explicit single-step method, the first-order Euler [17], and the fourth-order Runge-Kutta [29, 3, 13]. These schemes are not limited by transient velocities or complexity in the velocity fields. The single-step explicit method is computationally simple,

but of limited accuracy. The other two methods can attain a high degree of accuracy, but may require a large number of steps (and therefore computation time) to do so.

The semianalytical technique combines aspects of analytical and numerical methods. It makes use of an analytical solution to the integral within an individual spatial cell under the assumption of a lowest order Raviart-Thomas [25] velocity field: within each spatial cell, the x -component is continuous piecewise-linear in the x -direction and discontinuous piecewise-constant in the y -direction, and the y -component is the reverse.

Tracking is then conducted through one cell at a time. This idea was first presented by Pollock [22] for steady-state flow conditions.

For particle tracking, the semi-analytical method has been extended to nonsteady-state conditions [26], for streamline tracing the original method of Pollock is sufficient.

Pollock's equations are derived assuming orthogonal grid blocks, but very few real reservoirs model use such a strict Cartesian framework anymore. *Prévost et al.* [24] has extended Pollock's method to structured and unstructured irregular grids. The method uses a transformation of cellface fluxes in the physical space to corresponding fluxes on unit cubes in a reference space. Then Pollock's method is applied in the reference space, and using an isoparametric transformation, the streamline is transformed back to the physical space.

For streamline tracing in 2D we use the bilinear transformation (6.1) as the isoparametric transformation from reference space to the physical space, in 3D we use a trilinear transformation.

8.2 Pollock Tracing in 2D

We will give a short description of Pollock tracing in 2D for a unit square cell. See Appendix B for a more complete description. Consider Figure 8.1 where the velocities u_A , u_B , v_C and v_D are known, e.g. by a mapping (Section 8.7) from \mathcal{P} to \mathcal{R} . For steady-state flow these velocities are constants. The position of the particle within the cell is given by

$$\mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j}, \quad \mathbf{r}(0) = P_0 \quad (8.2)$$

and for steady flow the velocity of the particle is

$$\mathbf{v}(x, y) = \frac{d\mathbf{r}}{dt} = \frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} = u(x, y)\mathbf{i} + v(x, y)\mathbf{j}, \quad \mathbf{v}(P_0) = \mathbf{v}_0 \quad (8.3)$$

Next we assume a lowest order Raviart-Thomas velocity field within the cell

$$u(x, y) = u(x) = G_x x + u_A \quad (8.4a)$$

$$v(x, y) = v(y) = G_y y + v_C \quad (8.4b)$$

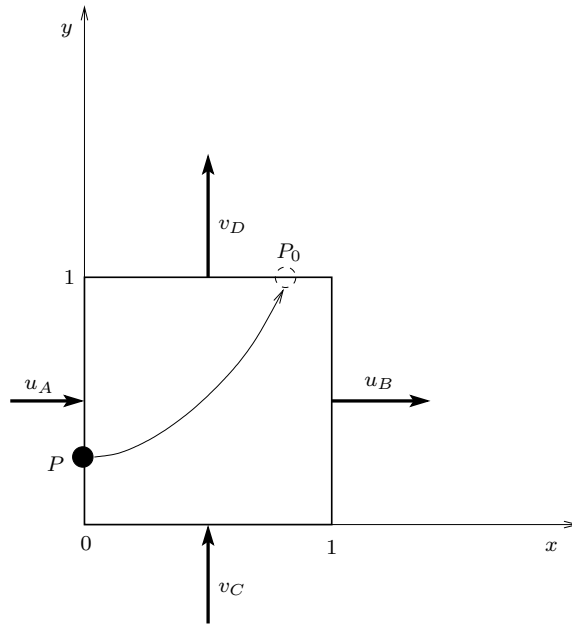


Figure 8.1: A unit cell in the reference space. A particle enters the cell at P_0 . Using Pollock's method, we can find the trajectory of the particle and its exit coordinate, P .

where $G_x = u_B - u_A$ and $G_y = v_D - v_C$.

Now (8.4a) gives that

$$\frac{dx}{dt} = G_x x + u_A \quad (8.5)$$

assuming $G_x \neq 0$ we can integrate this to give

$$t = \frac{1}{G_x} \ln \left(\frac{G_x x(t) + u_A}{u(0)} \right) \quad (8.6)$$

and this can be solved for $x(t)$ to give

$$x(t) = \frac{1}{G_x} (u(0)e^{G_x t} - u_A) \quad (8.7)$$

Similarly, from (8.4b) we get that

$$t = \frac{1}{G_y} \ln \left(\frac{G_y y(t) + v_C}{v(0)} \right) \quad (8.8)$$

and then

$$y(t) = \frac{1}{G_y} (v(0)e^{G_y t} - v_C) \quad (8.9)$$

Equation (8.7) and (8.9) represents the trajectory of the particle within the cell, where by (8.4), $u(0) = G_x x(0) + u_A$ and $v(0) = G_y y(0) + v_C$. The last step is then to find the exit point, P , or equivalently the travel time, T , used to traverse the cell.

Inserting $x(t) = 1$ in (8.6) we find the time, T_x , needed for the particle to reach the line $x = 1$, similarly, inserting $y(t) = 1$ in (8.8), we find the time, T_y , the particle uses to reach the line $y = 1$. Now the minimum, of T_x and T_y will give T , and then the exit coordinate is given by $(1, y_e)$ if $T = T_x$ or $(x_e, 1)$ if $T = T_y$. Here $x_e = x(T_y)$ and $y_e = y(T_x)$ as given by (8.6) and (8.8) respectively.

Note that the travel time, T , is the time-of-flight contribution, T_i , from the given cell. Thus the TOF for a given streamline can be calculated according to

$$\tau = \sum_i T_i \quad (8.10)$$

where i runs over all cells traversed by the given streamline.

8.3 Invariance of streamlines w.r.t. a scale of velocities

When tracing streamlines, we will keep two things in mind. First the shape of the streamlines depends only on the direction of the velocity field. Secondly the time-of-flight depends on the absolute value of the velocity field (and implicitly on the direction, since it must be calculated along a given streamline). The result in this section will be used later when we discuss how to improve the TOF for the Prévost-method.

Consider a velocity field in 2D at a certain time t_0 , $\mathbf{v}(x, y, t_0)$. At every point, $\mathbf{v}(x, y, t_0)$ is tangent to a given streamline. A scaling of $\mathbf{v}(x, y, t_0)$, say $k(x, y, t_0)\mathbf{v}(x, y, t_0)$, where $k(x, y, t_0) \neq 0$ is a scalar function, will not change the direction of these tangent vectors. See figure 8.2. Of course, such a scaling will give different travel times, but the shape of the streamline is the same.

Consider now the streamline tracing method described in the previous section. From (8.3) and (8.4) we have that the velocity in the unit cell is given by

$$\begin{aligned} \mathbf{v}(x, y) &= u(x)\mathbf{i} + v(y)\mathbf{j} = (G_x x + u_A)\mathbf{i} + (G_y y + v_C)\mathbf{j} = \\ &((u_B - u_A)x + u_A)\mathbf{i} + ((v_D - v_C)y + v_C)\mathbf{j} \end{aligned} \quad (8.11)$$

and take a *uniform* scaling of \mathbf{v} , i.e., $k(x, y)\mathbf{v} = k\mathbf{v}$, where k is a constant. Such a scaling can be done by setting $\tilde{u}_A = ku_A$, $\tilde{u}_B = ku_B$, $\tilde{v}_C = kv_C$ and

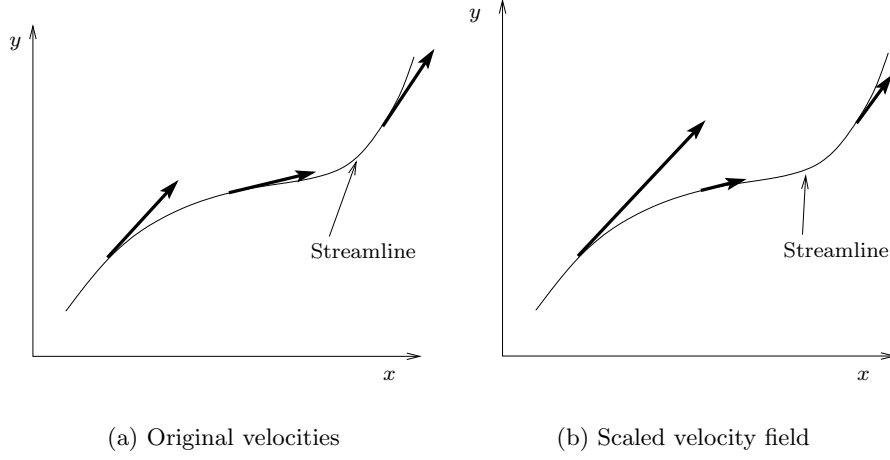


Figure 8.2: A scaling of velocities at time t_0 . The streamlines are still the same.

$\tilde{v}_D = kv_D$, because then,

$$\begin{aligned} \tilde{\mathbf{v}} &= ((\tilde{u}_B - \tilde{u}_A)x + \tilde{u}_A)\mathbf{i} + ((\tilde{v}_D - \tilde{v}_C)y + \tilde{v}_C)\mathbf{j} = \\ &= ((ku_B - ku_A)x + ku_A)\mathbf{i} + ((kv_D - kv_C)y + kv_C)\mathbf{j} = k\mathbf{v} \quad (8.12) \end{aligned}$$

where $\tilde{\mathbf{v}}$ is the scaled velocity.

This means that the shape of the streamline is invariant with respect to a uniform scaling of the cell-edge velocities.

8.4 Flux mappings

In this section we will consider a mapping of fluxes from \mathcal{P} to \mathcal{R} , and we establish some preliminary results.

Consider two adjacent quadrilaterals Q_1 and Q_2 of the grid. See Figure 8.3. We will define unit normals \mathbf{n}_{Ai} , \mathbf{n}_{Bi} , \mathbf{n}_{Ci} and \mathbf{n}_{Di} to the A -, B -, C and D -edges respectively, of a quadrilateral Q_i of the simulation grid. See Section 5.1 for notation.

Definition 8.1 (Normal vectors) *Given a quadrilateral, Q , in \mathcal{P} . For an A -edge or an C -edge we define the direction of the unit normal to be inwards relative to Q . For an B -edge or an D -edge we define the direction of the unit normal to be outwards relative to Q .*

We will also use the notation $\mathbf{n}_1 = \mathbf{n}_A$, $\mathbf{n}_2 = \mathbf{n}_B$, $\mathbf{n}_3 = \mathbf{n}_C$ and $\mathbf{n}_4 = \mathbf{n}_D$

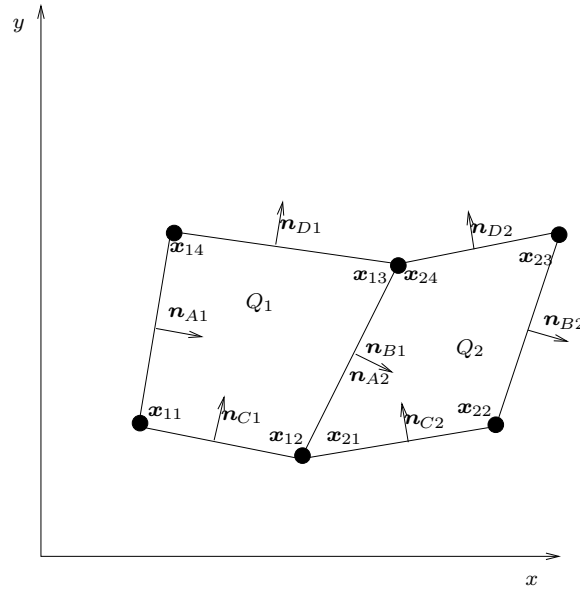


Figure 8.3: Two adjacent gridcells

Definition 8.2 (Fluxes) Given a velocity field \mathbf{q} on the edges of Q , we define the fluxes,

$$f_A = f_1 = \int_A \mathbf{q} \cdot \mathbf{n}_A ds \quad (8.13)$$

$$f_B = f_2 = \int_B \mathbf{q} \cdot \mathbf{n}_B ds \quad (8.14)$$

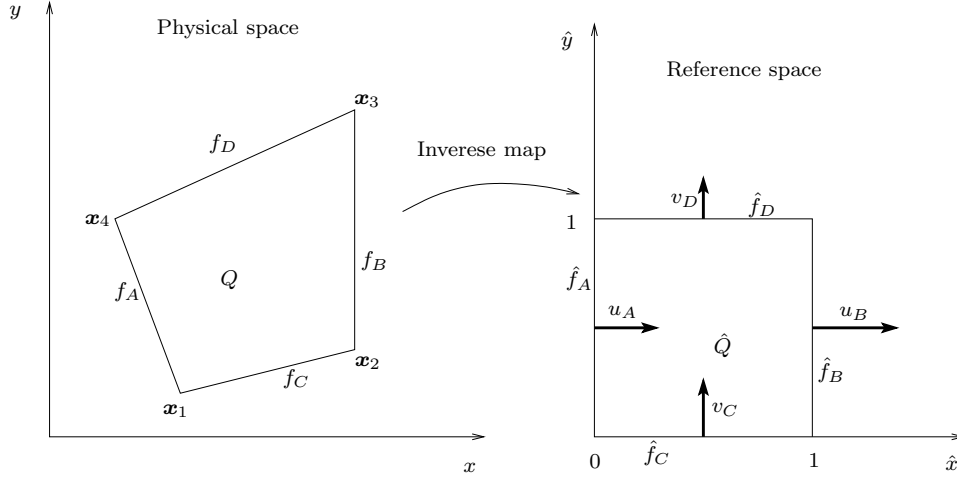
$$f_C = f_3 = \int_C \mathbf{q} \cdot \mathbf{n}_C ds \quad (8.15)$$

$$f_D = f_4 = \int_D \mathbf{q} \cdot \mathbf{n}_D ds \quad (8.16)$$

Observe that for Figure 8.3, $f_{B1} = f_{A2}$. Next consider an inverse bilinear map E of a quadrilateral Q in the physical space to a unit square \hat{Q} in a reference space. See Figure 8.4. By Section 6.2.1 the A -edge of Q gets mapped to an \hat{A} -edge of \hat{Q} .

Later in this chapter we will consider different flux mappings from \mathcal{P} to \mathcal{R} . Consider a mapping $\hat{f}_A = k f_A$ where k is positive constant, where the flux on the A -edge of Q gets mapped to a corresponding flux on \hat{A} -edge of \hat{Q} .

Then if f_A is positive, this must correspond to a flow into Q since \mathbf{n}_A point into Q . But a flow into \hat{Q} over the edge \hat{A} must correspond to a positive \hat{x} -component, u_A , of a corresponding velocity field $\hat{\mathbf{q}}$ in \mathcal{R} , since the \hat{A} -edge lies to left of the \hat{B} -edge on the \hat{x} -line.

Figure 8.4: Inverse map of Q

This argument shows that we can define edge velocities in the reference space corresponding this kind of transformed fluxes \hat{f}_i as,

$$\hat{q}_1 = u_A \hat{i} = \hat{f}_A \hat{i} \quad (8.17a)$$

$$\hat{q}_2 = u_B \hat{i} = \hat{f}_B \hat{i} \quad (8.17b)$$

$$\hat{q}_3 = v_C \hat{j} = \hat{f}_C \hat{j} \quad (8.17c)$$

$$\hat{q}_4 = v_D \hat{j} = \hat{f}_D \hat{j} \quad (8.17d)$$

since then

$$\hat{f}_i = \int_{\hat{E}_i} \hat{q}_i \cdot \hat{n}_i ds \quad (8.18)$$

See Figure 8.4.

8.5 Velocity mappings

In this section we will describe a direct method of mapping celledge velocities from \mathcal{P} to \mathcal{R} . This method is still under investigation, and will only be used for comparison to the derivation given by Prévost in [24] in the next section.

We will denote the velocity in \mathcal{P} by \mathbf{V} and the velocity in \mathcal{R} by $\hat{\mathbf{V}}$ such that

$$\mathbf{V} = \frac{d\mathbf{x}(\hat{\mathbf{x}})}{dt} = \frac{dx(\hat{\mathbf{x}})}{dt} \mathbf{i} + \frac{dy(\hat{\mathbf{x}})}{dt} \mathbf{j} \quad (8.19)$$

$$\hat{\mathbf{V}} = \frac{d\hat{\mathbf{x}}(\mathbf{x})}{dt} = \frac{d\hat{x}(\mathbf{x})}{dt} \hat{\mathbf{i}} + \frac{d\hat{y}(\mathbf{x})}{dt} \hat{\mathbf{j}} \quad (8.20)$$

where $\hat{\mathbf{i}} = \mathbf{i}$ and $\hat{\mathbf{j}} = \mathbf{j}$.

The relation between the velocity \mathbf{V} in \mathcal{P} and the velocity $\hat{\mathbf{V}}$ in \mathcal{R} is then by the chain rule and (6.14),

$$\hat{\mathbf{V}} = \frac{d\hat{\mathbf{x}}}{dt} = \frac{d\hat{\mathbf{x}}}{d\mathbf{x}} \frac{d\mathbf{x}}{dt} = D^{-1}\mathbf{V} = \frac{1}{J} \begin{bmatrix} \nu_1^T \\ \nu_2^T \end{bmatrix} \mathbf{V} = \hat{\mathbf{V}}_{\hat{x}} \hat{\mathbf{i}} + \hat{\mathbf{V}}_{\hat{y}} \hat{\mathbf{j}} \quad (8.21)$$

where

$$\hat{\mathbf{V}}_{\hat{x}} = \left(\frac{1}{J} \boldsymbol{\nu}_1 \cdot \mathbf{V} \right) \hat{\mathbf{i}} \quad (8.22a)$$

$$\hat{\mathbf{V}}_{\hat{y}} = \left(\frac{1}{J} \boldsymbol{\nu}_2 \cdot \mathbf{V} \right) \hat{\mathbf{j}} \quad (8.22b)$$

Since $\boldsymbol{\nu}_1 = (\frac{\partial y}{\partial \hat{y}}, -\frac{\partial x}{\partial \hat{y}})$ it follows that $\boldsymbol{\nu}_1$ is a constant vector along a \hat{x} -curve. Similarly $\boldsymbol{\nu}_2$ is a constant vector along a \hat{y} -curve.

Note that the length of the edge A in \mathcal{P} is given by (see (6.21))

$$\int_0^1 \left\| \frac{d\mathbf{x}}{d\theta} \right\| d\theta = \int_0^1 \left\| \frac{d\mathbf{x}(0, \hat{y})}{d\hat{y}} \right\| d\theta = \int_0^1 \left\| \boldsymbol{\nu}_1(0, \hat{y}) \right\| d\theta = \|\boldsymbol{\nu}_A\| \quad (8.23)$$

since $\boldsymbol{\nu}_1$ is a constant vector along a \hat{x} -curve. Here $\boldsymbol{\nu}_A$ is $\boldsymbol{\nu}_1$ evaluated at the edge A .

When we only know the flux, f_A , of \mathbf{V} along the edge A , we define the vector

$$\mathbf{V}_A = \frac{f_A}{\|\boldsymbol{\nu}_A\|} \frac{\boldsymbol{\nu}_A}{\|\boldsymbol{\nu}_A\|} \quad (8.24)$$

that corresponds to the flux f_A , since

$$\int_A \mathbf{V}_A \cdot \mathbf{n}_A ds = \int_A \frac{f_A}{\|\boldsymbol{\nu}_A\|} \frac{\boldsymbol{\nu}_A}{\|\boldsymbol{\nu}_A\|} \cdot \frac{\boldsymbol{\nu}_A}{\|\boldsymbol{\nu}_A\|} ds = \int_A \frac{f_A}{\|\boldsymbol{\nu}_A\|} ds = f_A \quad (8.25)$$

because the length of edge A is $\|\boldsymbol{\nu}_A\|$.

Next transform the vector \mathbf{V}_A to \mathcal{R} using (8.22),

$$\hat{\mathbf{V}}_{\hat{x}} = \frac{1}{J} \boldsymbol{\nu}_A \cdot \frac{f_A}{\|\boldsymbol{\nu}_A\|} \frac{\boldsymbol{\nu}_A}{\|\boldsymbol{\nu}_A\|} \hat{\mathbf{i}} = \frac{f_A}{J} \hat{\mathbf{i}} \quad (8.26a)$$

$$\hat{\mathbf{V}}_{\hat{y}} = \frac{1}{J} \boldsymbol{\nu}_B \cdot \frac{f_A}{\|\boldsymbol{\nu}_A\|} \frac{\boldsymbol{\nu}_A}{\|\boldsymbol{\nu}_A\|} \hat{\mathbf{j}} \quad (8.26b)$$

The \hat{y} -component, $\hat{\mathbf{V}}_{\hat{y}}$, in (8.26b) of the transformed velocity is generally not zero. We will use another expression for this component below, so for the moment, consider only the \hat{x} -component in (8.26a). Observe that this vector is not constant along the edge \hat{A} in \mathcal{R} because J varies along this edge. We will approximate J by J_A , where J_A is J evaluated at the midpoint of A , and we denote the resulting \hat{x} -component of the transformed \mathbf{V}_A , by $\hat{\mathbf{V}}_{\hat{A}}$. We do similar transformations for the other edges to obtain the transformed edge velocities,

$$\hat{\mathbf{V}}_{\hat{A}} = \frac{f_B}{J_A} \hat{\mathbf{i}} \quad \hat{\mathbf{V}}_{\hat{B}} = \frac{f_B}{J_B} \hat{\mathbf{i}} \quad \hat{\mathbf{V}}_{\hat{C}} = \frac{f_C}{J_C} \hat{\mathbf{j}} \quad \hat{\mathbf{V}}_{\hat{D}} = \frac{f_D}{J_D} \hat{\mathbf{j}} \quad (8.27)$$

if we again approximate the Jacobi determinants, J_A , J_B , J_C , and J_D by \tilde{J}_m , where J_m is the midpoint Jacobi determinant, this will give the approximate celledge velocities

$$\tilde{\mathbf{V}}_{\hat{A}} = \frac{f_B}{J_m} \hat{\mathbf{i}} \quad \tilde{\mathbf{V}}_{\hat{B}} = \frac{f_B}{J_m} \hat{\mathbf{i}} \quad \tilde{\mathbf{V}}_{\hat{C}} = \frac{f_C}{J_m} \hat{\mathbf{j}} \quad \tilde{\mathbf{V}}_{\hat{D}} = \frac{f_D}{J_m} \hat{\mathbf{j}} \quad (8.28)$$

We now define an approximate cell velocity in \hat{Q} ,

$$\begin{aligned} \tilde{\mathbf{V}} &= (\tilde{\mathbf{V}}_{\hat{B}} - \tilde{\mathbf{V}}_{\hat{A}})\hat{x} + \tilde{\mathbf{V}}_{\hat{A}} + (\tilde{\mathbf{V}}_{\hat{D}} - \tilde{\mathbf{V}}_{\hat{C}})\hat{y} + \tilde{\mathbf{V}}_{\hat{C}} = \\ &= \frac{1}{J_m} \left(((f_B - f_A)\hat{x} + f_A)\hat{\mathbf{i}} + ((f_D - f_C)\hat{y} + f_C)\hat{\mathbf{j}} \right) \end{aligned} \quad (8.29)$$

Thus we have approximated (8.26a) by $\frac{f_A}{J_m} \hat{\mathbf{i}}$ and (8.26b) by $\frac{(f_D - f_C)\hat{y} + f_C}{J_m} \hat{\mathbf{j}}$.

8.6 A consistent tracing velocity

In this section we will derive a consistent tracing velocity in \mathcal{R} for one phase incompressible flow. The derivation has been done for two phase horizontal flow in two dimensions in [24].

Consider incompressible single phase flow in 2D. Then ignoring diffusion and dispersion the conservation equation of a miscible solvent away from sources is given by

$$\frac{\partial C}{\partial t} + \nabla \cdot (C\mathbf{V}) = 0 \quad (8.30)$$

where $C = C(x, y, t)$ is the solvent concentration and \mathbf{V} is the total velocity. Gravitational effects are accomodated in the total velocity.

For incompressible flow we have

$$\nabla \cdot \mathbf{V} = 0 \quad (8.31)$$

This gives

$$\nabla \cdot (C\mathbf{V}) = C\nabla \cdot \mathbf{V} + \mathbf{V} \cdot \nabla C = \mathbf{V} \cdot \nabla C \quad (8.32)$$

so Equation (8.30) can be written

$$\frac{\partial C}{\partial t} + \mathbf{V} \cdot \nabla C = 0 \quad (8.33)$$

8.6.1 The Prévost derivation.

We will use the same derivation as in [24], the only difference is that Prévost considered two phase flow.

For a quadrilateral Q in \mathcal{P} we have using (8.31),

$$\int_Q \nabla \cdot \mathbf{V} dV = \int_{\partial Q} \mathbf{V} \cdot \mathbf{n} dS = \sum_{j=1}^4 \int_{E_j} \mathbf{V} \cdot \mathbf{n}_j dS = -f_1 + f_2 - f_3 + f_4 = 0 \quad (8.34)$$

since by Definition 8.1, \mathbf{n}_1 and \mathbf{n}_3 points into Q , whereas \mathbf{n}_2 and \mathbf{n}_4 points out of Q .

Equation (8.33) can be written

$$\frac{\partial C}{\partial t} + \mathbf{V} \cdot \nabla C = \frac{\partial C}{\partial t} + uC_x + vC_y = 0 \quad (8.35)$$

where we use $\mathbf{V} = (u, v)$. Then streamline tracing is performed with

$$\frac{dx}{dt} = u \quad (8.36)$$

$$\frac{dy}{dt} = v \quad (8.37)$$

Now integrating Equation (8.30) over a quadrilateral Q in physical space and using the divergence theorem, we get

$$\begin{aligned} \int_Q \frac{\partial C}{\partial t} dV + \int_{\partial Q} C\mathbf{V} \cdot \mathbf{n} dS = \\ \int_Q \frac{\partial C}{\partial t} dV + \sum_{j=1}^4 \int_{E_j} C\mathbf{V} \cdot \mathbf{n}_j dS = 0 \end{aligned} \quad (8.38)$$

We know change variables using the bilinear transformation (6.1), $\mathbf{x} = \mathbf{x}(\hat{\mathbf{x}})$. Then

$$\int_Q \frac{\partial C}{\partial t} dV = \int_{\hat{Q}} \frac{\partial C}{\partial t} J d\hat{V} \quad (8.39)$$

where we assume that J is positive. In Section 6.4 we showed that

$$\int_{E_1} C\mathbf{V} \cdot \mathbf{n}_1 dS = \int_{\hat{E}_1} C\mathbf{V} \cdot \boldsymbol{\nu}_1 d\hat{y} = \int_{\hat{Q}} \frac{\partial}{\partial \hat{x}} ((C\mathbf{V} \cdot \boldsymbol{\nu}_1)) d\hat{x} d\hat{y} \quad (8.40)$$

Similarly,

$$\int_{E_2} C\mathbf{V} \cdot \mathbf{n}_2 dS = \int_{\hat{Q}} \frac{\partial}{\partial \hat{x}} (C\mathbf{V} \cdot \boldsymbol{\nu}_1) d\hat{x} d\hat{y} \quad (8.41)$$

$$\int_{E_3} C\mathbf{V} \cdot \mathbf{n}_3 dS = \int_{\hat{Q}} \frac{\partial}{\partial \hat{y}} (C\mathbf{V} \cdot \boldsymbol{\nu}_2) d\hat{x} d\hat{y} \quad (8.42)$$

$$\int_{E_4} C\mathbf{V} \cdot \mathbf{n}_4 dS = \int_{\hat{Q}} \frac{\partial}{\partial \hat{y}} (C\mathbf{V} \cdot \boldsymbol{\nu}_2) d\hat{x} d\hat{y} \quad (8.43)$$

So (8.38) can be written

$$\int_{\hat{Q}} \left\{ \frac{\partial C}{\partial t} J + [C\tilde{F}]_{\hat{x}} + [C\tilde{G}]_{\hat{y}} \right\} d\hat{x} d\hat{y} = 0 \quad (8.44)$$

where $\tilde{F} = \mathbf{V} \cdot \boldsymbol{\nu}_1$ and $\tilde{G} = \mathbf{V} \cdot \boldsymbol{\nu}_2$. Now

$$[C\tilde{F}]_{\hat{x}} + [C\tilde{G}]_{\hat{y}} = [\tilde{F}C]_{\hat{x}} + [\tilde{G}C]_{\hat{y}} + C(\tilde{F}_{\hat{x}} + \tilde{G}_{\hat{y}}) \quad (8.45)$$

So the integrand of (8.44) can be put into the form

$$\left[\frac{\partial C}{\partial t} + \hat{u}C_{\hat{x}} + \hat{v}C_{\hat{y}} \right] J \quad (8.46)$$

where

$$\hat{u} = \frac{\tilde{F}}{J}, \quad \hat{v} = \frac{\tilde{G}}{J} \quad (8.47)$$

provided $C(\tilde{F}_{\hat{x}} + \tilde{G}_{\hat{y}})$ is zero. Now

$$\begin{aligned} \tilde{F}_{\hat{x}} &= \frac{\partial}{\partial \hat{x}}(\mathbf{V} \cdot \boldsymbol{\nu}_1) = \frac{\partial}{\partial \hat{x}} \left(u \frac{\partial y}{\partial \hat{y}} - v \frac{\partial x}{\partial \hat{y}} \right) = \\ &= \frac{\partial u}{\partial x} \frac{\partial x}{\partial \hat{x}} \frac{\partial y}{\partial \hat{y}} + u \frac{\partial^2 y}{\partial \hat{x} \partial \hat{y}} - \frac{\partial v}{\partial y} \frac{\partial y}{\partial \hat{x}} \frac{\partial x}{\partial \hat{y}} - v \frac{\partial^2 x}{\partial \hat{x} \partial \hat{y}} \end{aligned} \quad (8.48)$$

and

$$\begin{aligned} \tilde{G}_{\hat{y}} &= \frac{\partial}{\partial \hat{y}}(\mathbf{V} \cdot \boldsymbol{\nu}_2) = \frac{\partial}{\partial \hat{y}} \left(-u \frac{\partial y}{\partial \hat{x}} + v \frac{\partial x}{\partial \hat{x}} \right) = \\ &= -\frac{\partial u}{\partial x} \frac{\partial x}{\partial \hat{y}} \frac{\partial y}{\partial \hat{x}} - u \frac{\partial^2 y}{\partial \hat{y} \partial \hat{x}} + \frac{\partial v}{\partial y} \frac{\partial y}{\partial \hat{y}} \frac{\partial x}{\partial \hat{x}} + v \frac{\partial^2 x}{\partial \hat{y} \partial \hat{x}} \end{aligned} \quad (8.49)$$

Thus by (8.48), (8.49) and (8.31)

$$\tilde{F}_{\hat{x}} + \tilde{G}_{\hat{y}} = J \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = J(\nabla \cdot \mathbf{V}) = 0 \quad (8.50)$$

so (8.46) is valid, and (8.47) defines the “exact” tracing velocity for incompressible flow. Observe that (8.47) gives the same velocity as the direct method (8.22).

The discrete tracing velocity can now be deduced. From (8.50) we have

$$\int_0^1 \int_0^1 (\tilde{F}_{\hat{x}} + \tilde{G}_{\hat{y}}) d\hat{x} d\hat{y} = \int_0^1 \bar{F}_{\hat{x}} d\hat{x} + \int_0^1 \bar{G}_{\hat{y}} d\hat{y} = 0 \quad (8.51)$$

where

$$\bar{F} = \int_0^1 \tilde{F} d\hat{y}, \quad \bar{G} = \int_0^1 \tilde{G} d\hat{x} \quad (8.52)$$

are the mean values of \tilde{F} and \tilde{G} , which we can approximate by

$$F(\hat{x}) = f_1(1 - \hat{x}) + f_2\hat{x} \quad (8.53)$$

$$G(\hat{y}) = f_3(1 - \hat{y}) + f_4\hat{y} \quad (8.54)$$

since by (8.34)

$$F_{\hat{x}} + G_{\hat{y}} = -f_1 + f_2 - f_3 + f_4 = 0 \quad (8.55)$$

and thus (8.51) is satisfied. Then by (8.47) we can define a discrete tracing velocity as

$$\hat{u} = \frac{F(\hat{x})}{J}, \quad \hat{v} = \frac{G(\hat{y})}{J} \quad (8.56)$$

Comparing the last expression to (8.29), we see that the result is exactly the same if we use $J_m = J$ in (8.29).

8.7 The Prévost method

Now Prévost suggests the following tracing quality criteria:

- a) The velocity satisfies the mass conservation equation over the cell, and the discrete integral of divergence is zero for incompressible flow.
- b) The normal component of flux is continuous when crossing any face of the cell.
- c) The normal component of velocity is consistent with the flux computed by the numerical scheme.

In the previous section we showed that tracing in \mathcal{R} using (8.56) results in a divergence free velocity. The derivation of the tracing velocity started from Equation (8.30), so the given tracing velocity satisfies the mass conservation equation over the cell.

Then in order to be able to integrate (8.56) analytically Prévost suggests to replace J by J_m .

The Prévost-method thus maps fluxes from \mathcal{P} to \mathcal{R} according to

$$\hat{f}_i = f_i/J_m, \quad i = A, B, C, D \quad (8.57)$$

where J_m is the jacobian determinant of the midpoint of the cell in \mathcal{P} and is assumed to be positive, and A, B, C, D refers to corresponding edges under the transformation. Then streamline tracing is performed using (8.56), which is seen to be equivalent to Pollock tracing in \mathcal{R} .

Note that, by Section 8.3, the shape of the streamlines will be the same, for any constant approximation of J in (8.56), and we will show later that approximating J by J_m can give incorrect time-of-flight. Thus we can approximate J by \tilde{J} , where \tilde{J} is an approximation of J that will give better time-of-flight values, and still the shape of the streamlines will be the same. We will refer to this method as the modified Prévost-method.

8.8 Direct transformation of the flow equation

We are currently working on more direct method to give a consistent tracing velocity. This work has not yet been completed, but we will show the results found so far.

We want to transform Equation (8.33) from \mathcal{P} to \mathcal{R} in terms of the inverse transform of (6.1). So assume we can find the inverse transform

$$\hat{\mathbf{x}}(\mathbf{x}) = \begin{bmatrix} \hat{x}(x, y) \\ \hat{y}(x, y) \end{bmatrix} \quad (8.58)$$

Let

$$C = C(\mathbf{x}(\hat{\mathbf{x}}), t) = \hat{C}(\hat{\mathbf{x}}(\mathbf{x}), t) \quad (8.59)$$

$$\mathbf{V} = \frac{d\mathbf{x}(\hat{\mathbf{x}})}{dt} = \frac{dx(\hat{\mathbf{x}})}{dt} \mathbf{i} + \frac{dy(\hat{\mathbf{x}})}{dt} \mathbf{j} \quad (8.60)$$

$$\hat{\mathbf{V}} = \frac{d\hat{\mathbf{x}}(\mathbf{x})}{dt} = \frac{d\hat{x}(\mathbf{x})}{dt} \hat{\mathbf{i}} + \frac{d\hat{y}(\mathbf{x})}{dt} \hat{\mathbf{j}} \quad (8.61)$$

where $\hat{\mathbf{i}} = \mathbf{i}$ and $\hat{\mathbf{j}} = \mathbf{j}$. Then

$$\frac{\partial C}{\partial t} \rightarrow \frac{\partial \hat{C}}{\partial t} \quad (8.62)$$

because the time is the same in the two spaces (the arrow means the transformation from \mathcal{P} to \mathcal{R}).

By (6.8),

$$\mathbf{V} = \frac{d\mathbf{x}}{dt} = \frac{d\mathbf{x}}{d\hat{\mathbf{x}}} \frac{d\hat{\mathbf{x}}}{dt} = D\hat{\mathbf{V}} \quad (8.63)$$

where D is the jacobian matrix of the transformation, and

$$(\nabla C)^T = \frac{\partial C}{\partial \mathbf{x}} = \frac{\partial \hat{C}}{\partial \hat{\mathbf{x}}} = \frac{\partial \hat{C}}{\partial \hat{\mathbf{x}}} \frac{d\hat{\mathbf{x}}}{d\mathbf{x}} = (\hat{\nabla} \hat{C})^T D^{-1} \quad (8.64)$$

This means that

$$\nabla C = D^{-T} \hat{\nabla} \hat{C} \quad (8.65)$$

So Equation (8.33) in \mathcal{R} is

$$\begin{aligned} \frac{\partial \hat{C}}{\partial t} + (D\hat{\mathbf{V}}) \cdot (D^{-T} \hat{\nabla} \hat{C}) = \\ \frac{\partial \hat{C}}{\partial t} + \hat{\mathbf{V}}^T D^T D^{-T} \hat{\nabla} \hat{C} = \frac{\partial \hat{C}}{\partial t} + \hat{\mathbf{V}} \cdot \hat{\nabla} \hat{C} = 0 \end{aligned} \quad (8.66)$$

So we get exactly the same equation in \mathcal{R} . If we can show that $\hat{\nabla} \cdot \hat{\mathbf{V}} = 0$, then it is believed that this is good starting point for deriving a consistent tracing velocity.

8.9 Tracing on irregular grids

We have now described the principals behind the Prévost-method. How this method is used in practice is described in Appendix C.

Chapter 9

The Piola Transformation

In order to show that the Prévost method reproduces streamlines of uniform flow, we will use a result from Russell in [19]. In this paper it is shown that the Piola transformation of a given lowest order Raviart-Thomas [25] velocity reproduces uniform flow.

The Piola transformation is a vector transformation from \mathcal{R} to \mathcal{P} that conserves fluxes over any \hat{x} -curve or \hat{y} -curve (see Section 9.2). The transformation is given by

$$\mathbf{v}(\mathbf{x}(\hat{\mathbf{x}})) = \frac{D(\hat{\mathbf{x}})}{J(\hat{\mathbf{x}})} \hat{\mathbf{v}}(\hat{\mathbf{x}}) \quad (9.1)$$

Here $\mathbf{v}(\mathbf{x}(\hat{\mathbf{x}}))$ is the vector in \mathcal{P} , $\hat{\mathbf{v}}(\hat{\mathbf{x}})$ is the vector in \mathcal{R} , $\mathbf{x}(\hat{\mathbf{x}})$ is the bilinear transformation, (6.1), $D(\hat{\mathbf{x}})$ is the jacobi matrix, (6.8), and $J(\hat{\mathbf{x}})$ is the jacobi determinant given in (6.9).

9.1 The inverse transformation

Assuming that the transformation $\mathbf{x}(\hat{\mathbf{x}})$ is invertible we can solve (9.1) for $\hat{\mathbf{v}}(\hat{\mathbf{x}})$, to give

$$\hat{\mathbf{v}}(\hat{\mathbf{x}}(\mathbf{x})) = J(\mathbf{x})D^{-1}(\mathbf{x})\mathbf{v}(\mathbf{x}) \quad (9.2)$$

9.2 Flux conservation

In Section 6.4, we showed that the flux of a vector field \mathbf{q} in \mathcal{P} over an \hat{x} -curve is given by

$$f_{\hat{x}} = \int_0^1 (\mathbf{q} \cdot \boldsymbol{\nu}_1)(\mathbf{x}(\hat{\mathbf{r}}(\theta)))d\theta \quad (9.3)$$

where $\boldsymbol{\nu}_1$ is a normal vector to the \hat{x} -curve, $\hat{\mathbf{r}}(\theta)$ is a representation for the \hat{x} -curve in \mathcal{R} and $\mathbf{x}(\cdot)$ is the bilinear transformation.

By transforming the vector \mathbf{q} to a vector $\hat{\mathbf{q}}$ in reference space, we want the corresponding flux in the two spaces to be equal. The corresponding

flux in reference space is given by

$$\int_{\hat{C}} (\hat{\mathbf{q}} \cdot \hat{\mathbf{N}})(\hat{\mathbf{r}}(\theta)) \left\| \frac{d\hat{\mathbf{r}}}{d\theta} \right\| d\theta \quad (9.4)$$

where $\hat{\mathbf{N}}$ is the unit normal to the curve \hat{C} . For the \hat{x} -curve given in (6.20), $\hat{\mathbf{N}} = [1, 0]$ and $\left\| \frac{d\hat{\mathbf{r}}}{d\theta} \right\| = 1$. So (9.4) becomes

$$\int_0^1 (\hat{\mathbf{q}} \cdot [1, 0])(\hat{\mathbf{r}}(\theta)) d\theta = \int_0^1 \hat{q}_1(\hat{\mathbf{r}}(\theta)) d\theta \quad (9.5)$$

Comparing (9.3) and (9.5) we want

$$\int_0^1 (\mathbf{q} \cdot \boldsymbol{\nu}_1)(\mathbf{x}(\hat{\mathbf{r}}(\theta))) d\theta = \int_0^1 \hat{q}_1(\hat{\mathbf{r}}(\theta)) d\theta \quad (9.6)$$

We can achieve this by setting

$$\hat{q}_1(\hat{\mathbf{r}}(\theta)) = (\mathbf{q} \cdot \boldsymbol{\nu}_1)(\mathbf{x}(\hat{\mathbf{r}}(\theta))) \quad (9.7)$$

Observe that $\hat{q}_2(\cdot)$ can be chosen without constraint in this case.

Similarly, for an \hat{y} -curve, we can achieve flux conservation by setting

$$\hat{q}_2(\hat{\mathbf{s}}(\theta)) = (\mathbf{q} \cdot \boldsymbol{\nu}_2)(\mathbf{x}(\hat{\mathbf{s}}(\theta))) \quad (9.8)$$

where $\hat{\mathbf{s}}(\theta)$ is a representation for a \hat{y} -curve. We see that there is no constraint on $\hat{q}_1(\cdot)$ for this case. Thus by (6.14) if we take

$$\hat{\mathbf{q}} = \begin{bmatrix} \boldsymbol{\nu}_1^T \\ \boldsymbol{\nu}_2^T \end{bmatrix} \mathbf{q} = J\tilde{D}^{-1}\mathbf{q} \quad (9.9)$$

we achieve flux conservation for both \hat{x} - and \hat{y} -curves. Comparing the last expression to (9.2), we see that $\hat{\mathbf{q}}$ is then the inverse Piola transform of \mathbf{q} .

9.3 Reproduction of uniform flow

Consider the transformation (8.21) of velocity from \mathcal{P} to \mathcal{R}

$$\hat{\mathbf{V}} = \frac{d\hat{\mathbf{x}}}{dt} = \frac{d\hat{\mathbf{x}}}{d\mathbf{x}} \frac{d\mathbf{x}}{dt} = D^{-1}\mathbf{V} \quad (9.10)$$

and compare it with the inverse Piola transformation

$$\hat{\mathbf{V}} = JD^{-1}\mathbf{V} \quad (9.11)$$

Thus the Piola transformation does not preserve the velocity if we use the same time coordinate in \mathcal{P} and \mathcal{R} .

Since the Piola transform conserves fluxes the inverse Piola transform can be seen as a flux mapping (Section 8.4) from \mathcal{P} to \mathcal{R} on the form

$$\hat{f}_i = f_i, \quad i = A, B, C, D \quad (9.12)$$

The standard basis functions in \mathcal{R} associated with the edgefluxes, f_i , are the lowest-order Raviart-Thomas (RT_0) functions [19, 25].

An RT_0 function in \mathcal{R} is a linear combination of the velocity fields \mathbf{u}_A , $\mathbf{u}_B, \mathbf{v}_C, \mathbf{v}_D$, given by

$$\mathbf{u}_A = (1 - \hat{x}, 0) \quad (9.13)$$

$$\mathbf{u}_B = (\hat{x}, 0) \quad (9.14)$$

$$\mathbf{v}_C = (0, 1 - \hat{y}) \quad (9.15)$$

$$\mathbf{v}_D = (0, \hat{y}) \quad (9.16)$$

It follows that the velocity field used in the Prévost-method is an RT_0 function.

In [19] it is shown that a uniform velocity field, \mathbf{q} , in \mathcal{P} , is the exact Piola transform of an RT_0 function.

Comparing (9.12) with the Prévost-method (Section 8.7) we see that the only difference is a different scaling of the fluxes in \mathcal{R} , thus by the discussion in 8.3, the shape of the streamlines of either method will be the same, and by the above argument, reproduce a uniform flow field. That is, the method will reproduce the shape of the streamlines for a uniform flow field, the TOF is not reproduced.

Chapter 10

Results

In this chapter we will look at some flow fields where the analytical solution is known and compare it with the results from the method. We will be interested in both the shape of the streamlines and the time-of-flight. First we will see that a uniform flow field is preserved, then we look at a circular velocity field. The next section tests a flow field with a divergence different from zero. We then examine the sensitivity with respect to perturbations of the grid. Then the problem with time-of-flight is considered, and at last we use streamlines to validate the results of the MPFA-method.

10.1 Uniform velocity field

For a uniform velocity field $\mathbf{q} = (q_x, q_y)$, where q_x and q_y are constants, we know the analytical solution: The streamlines should be straight lines with slope q_y/q_x . For such a velocity field we can calculate the edge fluxes analytically (Appendix G.1), and this can be used as input to the tracing algorithm. The streamlines of the method can then be compared to the exact solution.

So assume the fluxes of each edge is given by (G.3), we then trace the streamlines in \mathcal{R} using the Prévost-method and map the solution back to \mathcal{P} using the bilinear transformation. Since the edge fluxes are given analytically, the other steps can also be done analytically (see Appendix G.2).

In the Appendix we showed that if $\mathbf{q} = (1, 0)$ the streamlines are reproduced exactly. We could not show this generally using this direct method, so we will rely on the result in Section 9.3.

The test will be performed on the 19x19 gridcell grid, \mathcal{G} , shown in Figure 10.1. The grid is constructed as described in (Appendix K) with $r_x = r_y = 0.6$. We will use 8 different flow fields, $\mathbf{q}_i, 1 \leq i \leq 8$. Such that $\mathbf{q}_i = (1, \tan((i * 10/180) * \pi))$, see Figure 10.2.

For each flow field we calculate the fluxes of \mathcal{G} analytically. We then generate one streamline for each flow field. Confer Appendix C, for a description

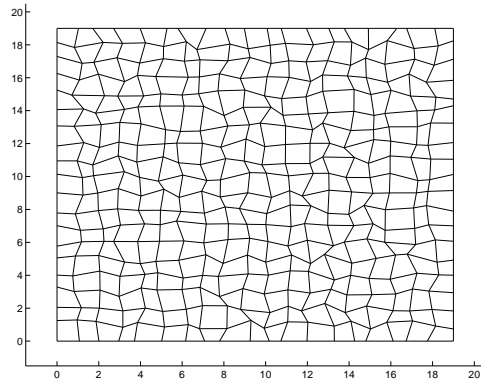
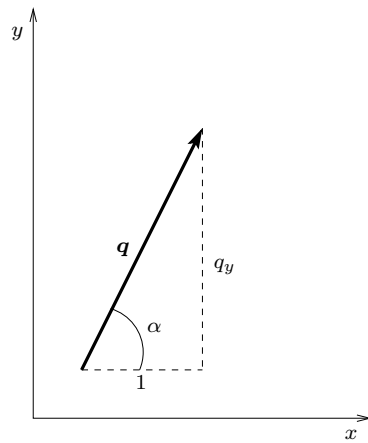


Figure 10.1: Grid that will be used to test a uniform flow.

Figure 10.2: The vector $\mathbf{q} = (1, q_y) = (1, \tan(\alpha))$.

of the tracing algorithm.

10.1.1 Shape of streamlines

The result is shown in Figure 10.3. The circles are starting points for the

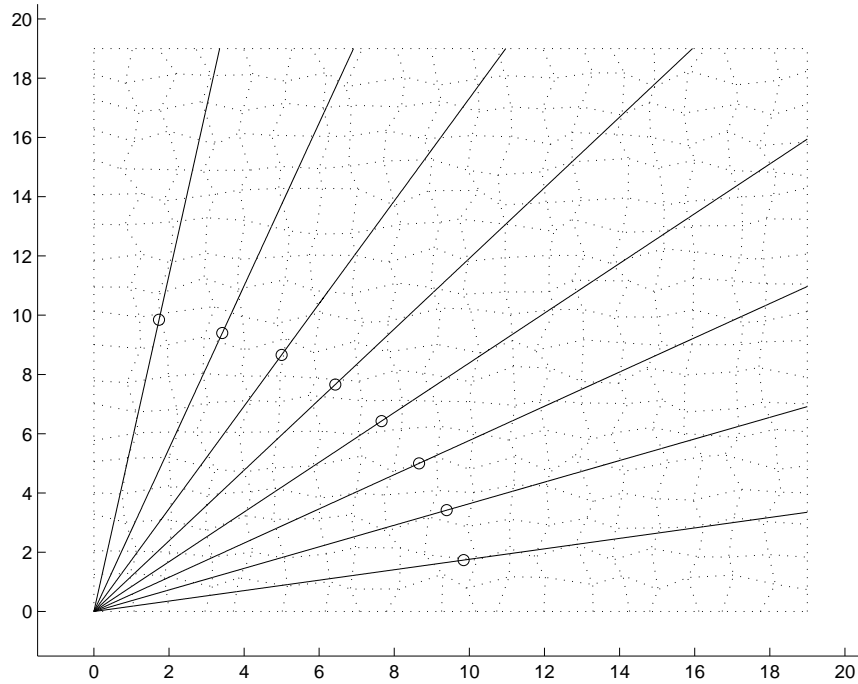


Figure 10.3: Each streamline correspond to a different uniform flow field. The circles are starting points for the streamlines.

different streamlines, and they all lie on a circle of radius 10 with center in $(0,0)$. It seems that the streamline method reproduces uniform flow streamlines exactly. This is also expected from the result in Section 9.3.

10.1.2 Time-of-flight

We also want to check the Time-of-flight calculations for the numerical method. The numerical TOF is calculated according to (8.10), the true TOF calculations is given in Appendix G.3.

The results are shown in Figure 10.4. We observe that for $\alpha = 30^\circ$ the

α	10	20	30	40	50	60	70	80
True TOF	19	19	19	19	15.9429	10.9697	6.9154	3.3502
Numerical TOF	19.0331	19.0078	19.6992	19.1104	15.9598	11.0094	6.8082	3.3685

Figure 10.4: TOF calculations for the uniform velocity field.

TOF is almost 4% in error.

10.2 Circular velocity field

Next we will test the tracing algorithm for a circular steady velocity field $\mathbf{q}(x, y)$. We know that the true streamlines should be circles. Using numerical integration we can define fluxes on each edge of the grid corresponding to \mathbf{q} . We then apply the tracing algorithm for these fluxes to see if it can match the true solution.

Given an origin $O = (x_0, y_0)$ and a point $P = (x, y)$, we can define a radius vector pointing from O to P as $\mathbf{r}(x, y) = [x - x_0, y - y_0]$. Consider a circle drawn with center in O and with radius $|\mathbf{r}|$. See Figure 10.5. Then

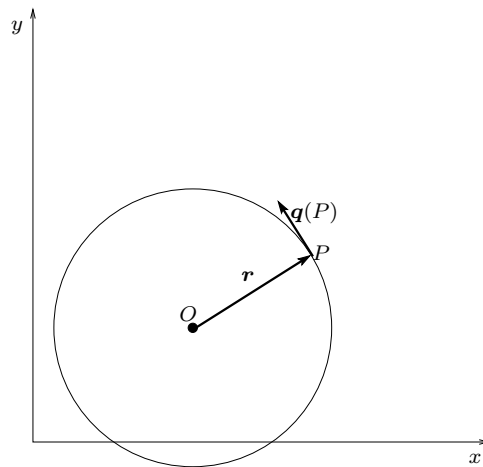


Figure 10.5: Definition of a circular velocity field

we define $\mathbf{q}(x, y)$ to be the unit tangent vector, at P , to the circle, pointing in the counter clockwise direction. Set $\mathbf{r}(x, y) = [r_1(x, y), r_2(x, y)]$, then we have that

$$\mathbf{q}(x, y) = \frac{1}{\|\mathbf{r}\|}[-r_2(x, y), r_1(x, y)] = \frac{1}{\sqrt{(x - x_0)^2 + (y - y_0)^2}}[y_0 - y, x - x_0] \quad (10.1)$$

since $\mathbf{q} \cdot \mathbf{r} = 0$ and $\|\mathbf{q}\| = (1/\|\mathbf{r}\|)\|\mathbf{r}\| = 1$. The vector \mathbf{q} points in counter clockwise direction since it can be thought of as a 90° counter clockwise rotation of \mathbf{r} . Such a rotation can be written as a linear transformation with matrix

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (10.2)$$

Then we see that $A\mathbf{r} = [-r_2, r_1]$.

We will use the grid, \mathcal{G} , shown in Figure 10.6. The grid is constructed

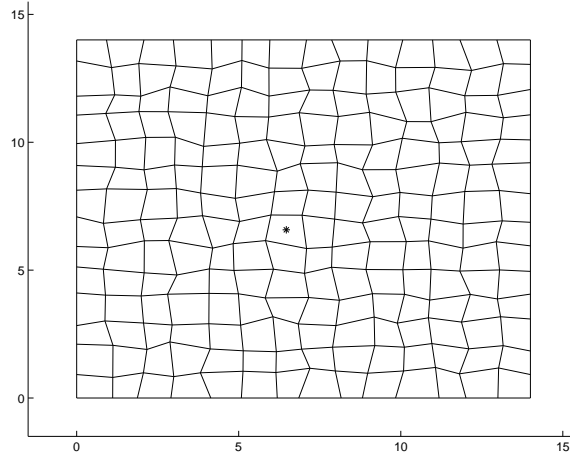


Figure 10.6: Grid to be used in the test case. The point marked with an asterisk will be the origin for the circular velocity field.

as described in (Appendix K) with $r_x = r_y = 0.4$, and is a “random” perturbation of a 14x14 gridcell square grid.

We calculate the fluxes numerically (Appendix H.1) for the velocity field described above.

10.2.1 Shape of streamlines

We select seven different starting points for streamlines within the grid. The result is shown in Figure 10.7. Note that for the four innermost circles, if we trace the streamlines forward from the circle point the streamline will end up at same point (or almost the same point) after passing through a given number of cells. This means that we have to give a stopping criteria to avoid an infinite loop. For this testcase we have chosen to stop tracing when the streamline has passed more than 80 gridcells.

Consider the innermost circle. One orbit for this circle contains 8 gridcells; this means that the circle is traced 10 times. It is surprising that from the figure, we cannot separate the ten circles. This shows a kind of stability of the method.

We note that the innermost circle is clearly deformed, so the shape of a circle is not preserved. This error can come from the computation of fluxes, or from the streamline method. We believe this error come from the tracing method, and shows that when the curvature of streamlines is big the error in the shape of the streamlines will grow. This point is to be investigated further.

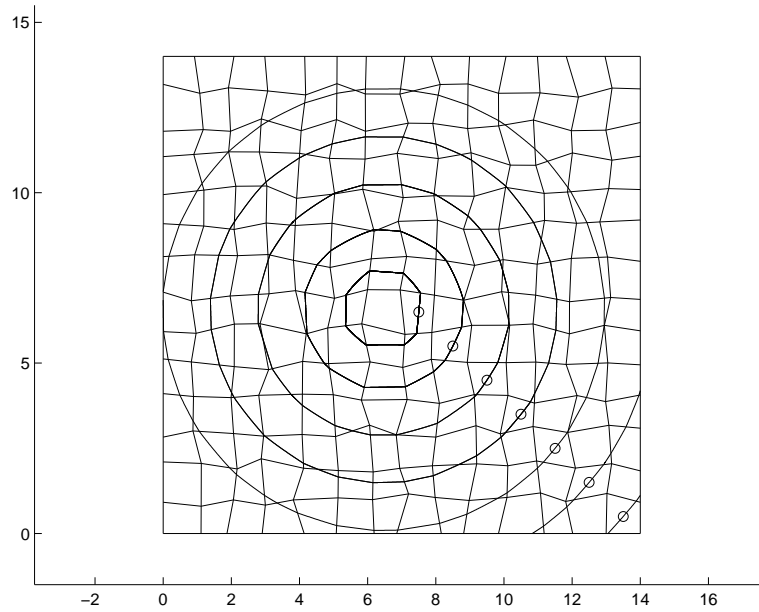


Figure 10.7: Grid with streamlines. The circles are startingpoints for a given streamline.

10.2.2 Time-of-flight

We will investigate the time-of-flight for the four innermost circles. See Figure 10.8. From (7.2), the TOF for a particle tracing one orbit of a circle,

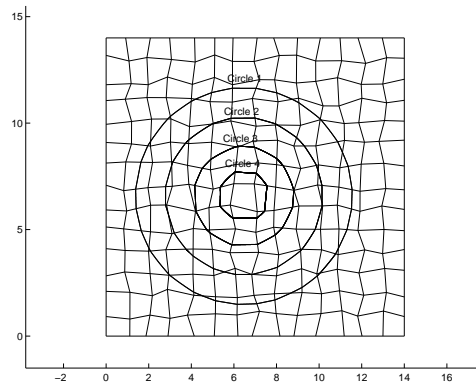


Figure 10.8: The four innermost circles.

is given by

$$\tau = \int_C \frac{1}{|\mathbf{q}|} ds = \int_C ds = \{\text{Circumference of circle}\} \quad (10.3)$$

since $|\mathbf{q}| = 1$.

For circle 1, we will consider 2 orbits, and the analytical TOF should then be $\tau_1 = 2 * 2\pi r \approx 4\pi * 5.071 \approx 63.72$, whereas the numerical method gives $\tilde{\tau}_1 = 63.51$.

For circle 2, the true TOF for 2 orbits is $\tau_2 = 46.17$, and the numerical result is $\tilde{\tau}_2 = 46.08$. For circle 3, the TOF's for 4 orbits is $\tau_3 = 57.78$ and $\tilde{\tau}_3 = 58.62$. For circle 4, with 10 orbits, we get, $\tau_4 = 65.04$ and $\tilde{\tau}_4 = 78.78$.

Thus there is a big error in TOF for the innermost circle. This is clearly something that should be investigated closer.

10.3 An y -dependent velocity field

We will test streamline tracing for a steady velocity field given by

$$\mathbf{q}(y) = \mathbf{i} + y\mathbf{j} \quad (10.4)$$

Note that $\nabla \cdot \mathbf{q} = 1$. In Section 8.6, we derived a consistent tracing velocity assuming $\nabla \cdot \mathbf{q} = 0$. Thus we don't know if the tracing velocity is consistent, for this case.

According to (4.7a) the streamlines are given by

$$\frac{dy}{dx} = \frac{q_y}{q_x} = y, \quad y(0) = y_0 \quad (10.5)$$

We integrate this to obtain

$$y(x) = y_0 e^x \quad (10.6)$$

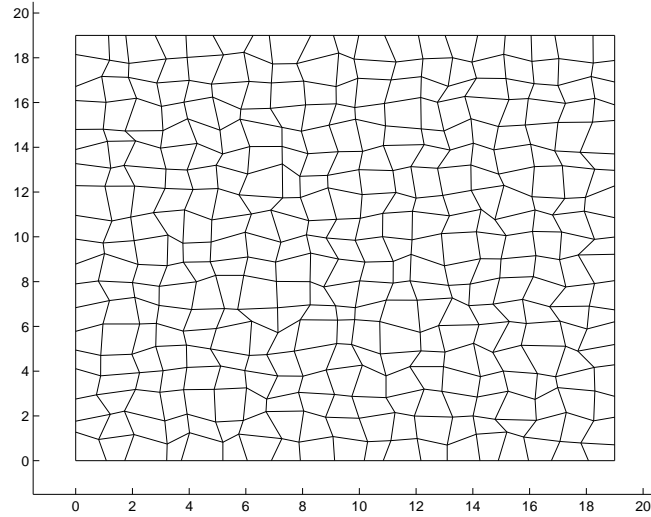
We will use a 19x19 gridcell random convex grid, \mathcal{G} , (Appendix K). Some analytical streamlines for this grid are shown in Figure 10.9.

The numerical streamlines are shown in Figure Figure 10.10(b) shows that the numerical method gives very good results for this test case.

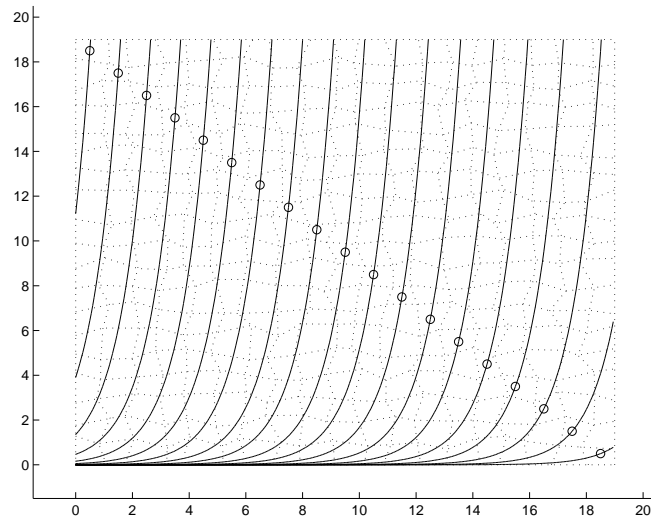
10.4 Sensitivity to perturbation of gridcells

In this section we will investigate how the shape of a streamline is affected with respect to perturbations of the grid. We will use a 20x20 gridcell grid. The flow problem will be single-phase incompressible horizontal flow in 2D for a homogeneous medium with no flow boundaries. There will be a source in lower right corner and a sink in the upper right corner, both of unit strength. The edgefluxes for the gridcells will be computed by the method described in Section 3.2.

We will use ten differently shaped grids, all 20x20 cells. The first grid is shown in Figure 10.11(a). Grid number two is obtained from the first by a small perturbation of all the cells in the first grid except for the cells containing the source and the sink (cell number 0 and cell number 399). The perturbation is done both uniformly and randomly.

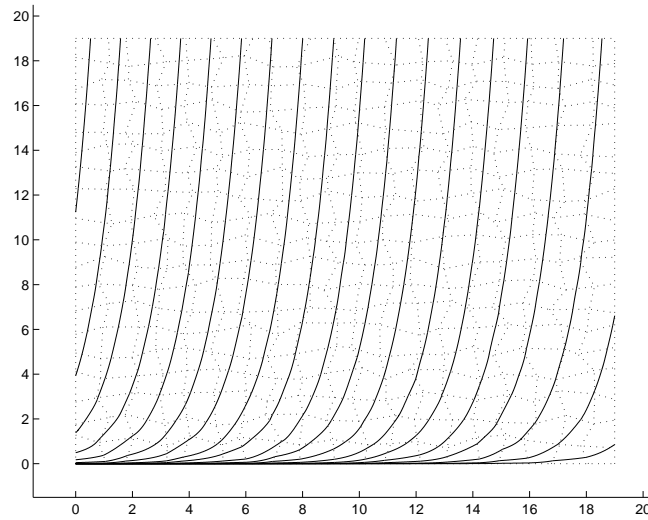


(a) The grid used in the test.

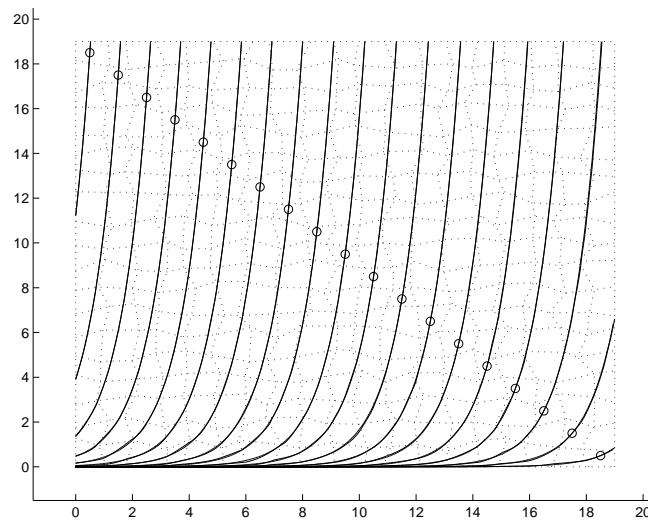


(b) Some analytical streamlines for the grid.

Figure 10.9: Streamlines for $\mathbf{q} = (1, y)$.

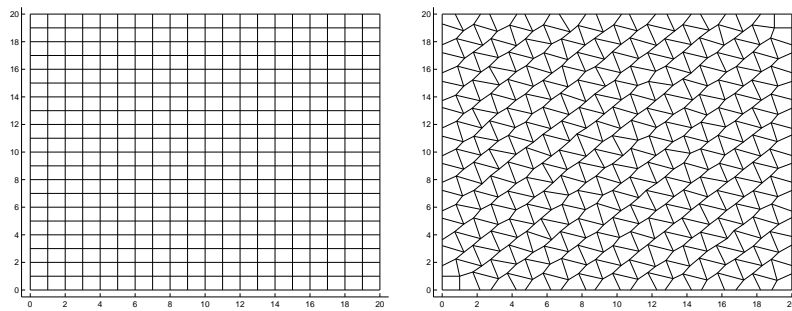


(a) Numerical streamlines.



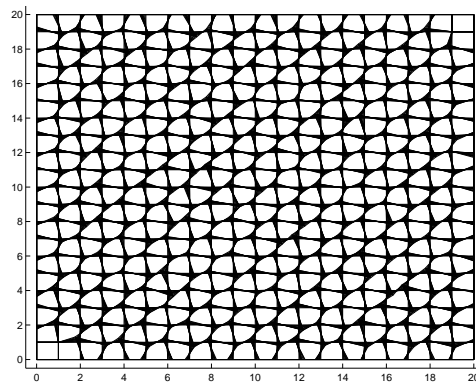
(b) Analytical and numerical in the same plot.

Figure 10.10: Streamlines for $\mathbf{q} = (1, y)$.



(a) The first grid.

(b) The last grid (i.e. Grid number 10).



(c) All the 10 grids plotted on top of each other.

Figure 10.11: The grids used in the test.

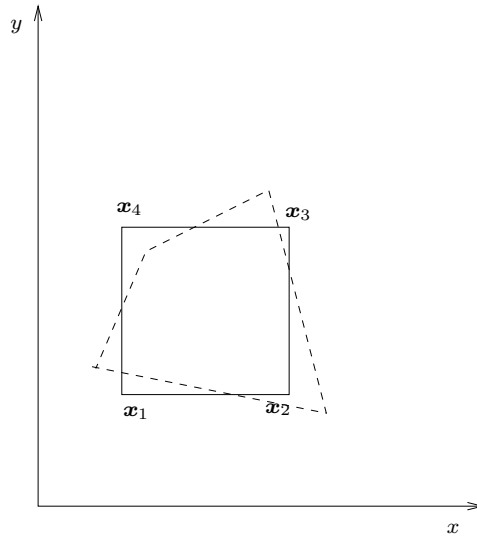


Figure 10.12: The dashed quadrilateral is obtained by a perturbation of the corners of the square quadrilateral.

For a cell not on the boundary of the solution domain, each gridpoint of the cell is given a perturbation like the one shown in figure 10.12. The perturbation of $\mathbf{x}_1 = (x_1, y_1)$ is done by subtracting a random number between 0 and 0.05 from x_1 , then adding a new random number between 0 and 0.05 to y_1 . Similar perturbations goes for the other corners.

The cells on the bottom and top boundary of the domain are not perturbed in the y -direction for the corners lying on the boundary, whereas the cells on the left and right boundary of the domain are not perturbed in the x -direction for the boundary cornerpoints. In this way the solution domain is kept fixed, and the same goes for the source and sink cells of the grids.

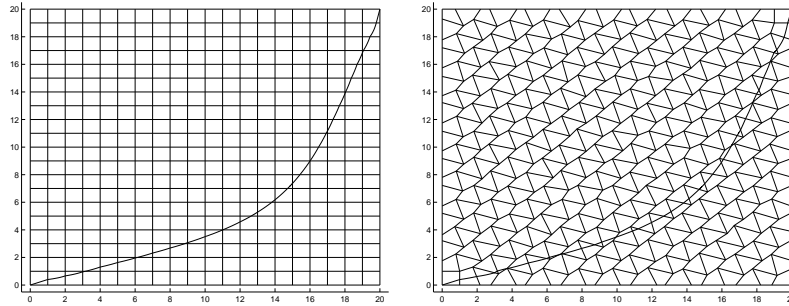
The other grids, grid number 3 to 10 is obtained from its previous grid in a similar fashion. Then the grids will be sequence of more and more “irregular” grids. See Figure 10.11. The reason for doing a random perturbation was to avoid cancellation effects when perturbing the grid by the same amount for each gridpoint.

We trace one streamline for each grid. Each streamline will be initiated from the point $(x, y) = (1.5, 0.5)$. We will assume that the MPFA-method gives correct fluxes for each grid. Then, in theory, all the streamlines should be identical, but due to errors introduced by the tracing algorithm there will be differences.

In Figure 10.13 we have plotted all the 10 streamlines in the first grid. We note that the method seems to be stable for this kind of perturbations.

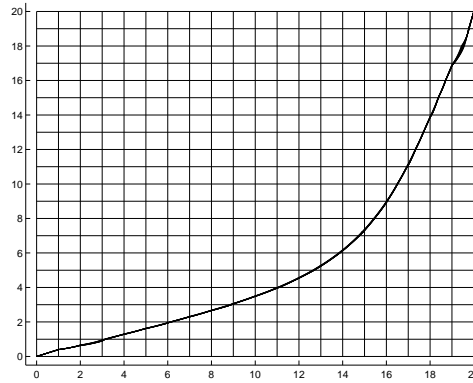
This test is to be extended to a case where we can calculate the fluxes analytically. We then don’t have to rely on a numerical method to compute

the fluxes.



(a) Streamline for the first grid.

(b) Streamline for the last grid.



(c) All the 10 streamlines plotted on top of each other in the first grid.

Figure 10.13: Streamlines for the testcase.

10.5 Time-of-flight

In this section we will show that the Prévost-method (Section 8.7) of flux mapping from \mathcal{P} to \mathcal{R} can give incorrect time-of-flight.

10.5.1 The single cell

We will first consider a simple testcase that will demonstrate the problem. See Figure 10.14(a). Assume a uniform velocity field, $\mathbf{q} = (1, 0)$ in \mathcal{P} . See Figure 10.14(b). Then a particle starting at P_1 should use $t = 1$ second to reach P_2 . But if we use the Prévost-method we will get $t = 1 + \beta$. See Appendix J for calculations. This is because the method uses the midpoint

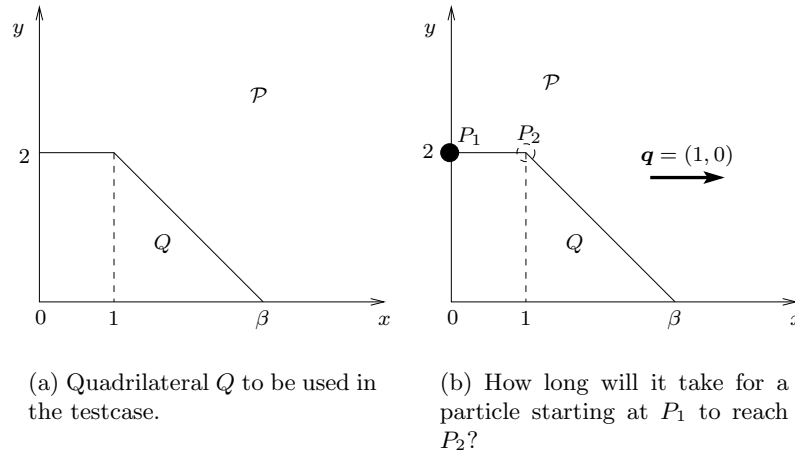


Figure 10.14: The problem to be investigated in the testcase.

Jacobi determinant, J_m , as a scaling factor for the fluxes (see Section 8.7). The midpoint Jacobian cannot represent the “stretching” of the spatial coordinates on this particular trajectory.

For Q we have

$$J = 2\hat{y}(1 - \beta) + 2\beta \tag{10.7}$$

Thus we see that $J_m = J(0.5, 0.5) = 1 + \beta$, whereas the Jacobi determinant, J_s , on the midpoint of the streamline is $J_s = J(0.5, 1) = 2$. If we use J_s in the tracing algorithm, we get $t = 1$ as we should.

10.5.2 Accumulation error in a skew grid

We next consider how this problem can accumulate for certain grids. See Figure 10.15. Each row of the grid contains trapezoids of equal area. See Appendix J for calculations. The particle p_i , $i = 1, 2, 3, 4$ starts at the point P_i at $t = 0$. The velocity field is uniform given by $\mathbf{q} = (1, 0)$, and the tracing is done by the Prévost-method.

Due to incorrect computations of the Jacobi-determinant, as described in the previous section, the particles will reach the vertical line, l , at $x = 20$ at different times. Let T_i be the time particle p_i arrives at l . First assume that p_1 and p_2 belongs to the first row of the grid (p_2 could also belong to the second row, since its trajectory lies on the interface between cells in the first row and the second row). Assume also that p_3 and p_4 belongs to the top row of the grid.

Then $T_1 = 19.75$, $T_2 = 20.25$, $T_3 = 19.48$ and $T_4 = 20.5$, whereas the correct time should be $T = 20$.

As shown in Appendix J, the time-of-flight is also discontinuous along the trajectories of p_2 and p_3 .

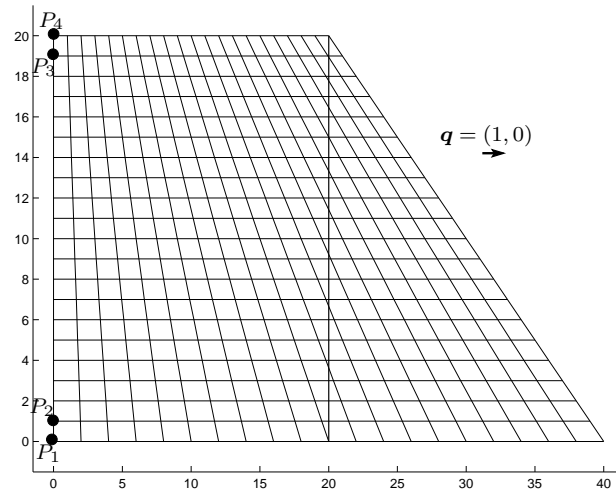


Figure 10.15: Consider four particles starting at P_1 , P_2 , P_3 and P_4 , respectively in a uniform flow field $\mathbf{q} = (1, 0)$. Which particle will reach the line $x = 20$ first?

10.6 Parallelograms and the MPFA-method

In this section we will use streamlines to investigate monotonicity for the MPFA-method. The MPFA-method [1] is a method for solving the pressure equation (3.7). For skewed parallelograms this method can give incorrect edgefluxes [20]. Using streamlines, we will try to visualize the problems, and determine when they arise.

We will still use a simple 20x20 gridcell grid. The flow problem will be single-phase incompressible horizontal flow in 2D for a homogeneous medium with no flow boundaries. There will be a source in lower right corner and a sink in the upper right corner, both of unit strength. The edgefluxes for the gridcells will be computed by the MPFA-method.

The grids will consist of mostly parallelograms of the type shown in Figure 10.16. The x -coordinates of the corners of the parallelograms are fixed such that we always have $x_2 - x_1 = 1$ and $x_3 - x_4 = 1$ (here as usual $\mathbf{x}_i = (x_i, y_i)$).

A grid of parallelograms, squares and trapezoids is then constructed, as shown in Figure 10.17. The grid in figure 10.17(a) has some large trapezoidal cells in second row from below. We would like to avoid having such cells in the grid, but in order to keep the boundary of the solution domain fixed for each testcase we had to use these cells. Still, we hope that these cells will not affect the solution in the middle of the domain, which is of the interest for this case.

The reason for the row of squares at the bottom and at the top of the grids, is too keep the source cell and the sink cell fixed, so the streamlines

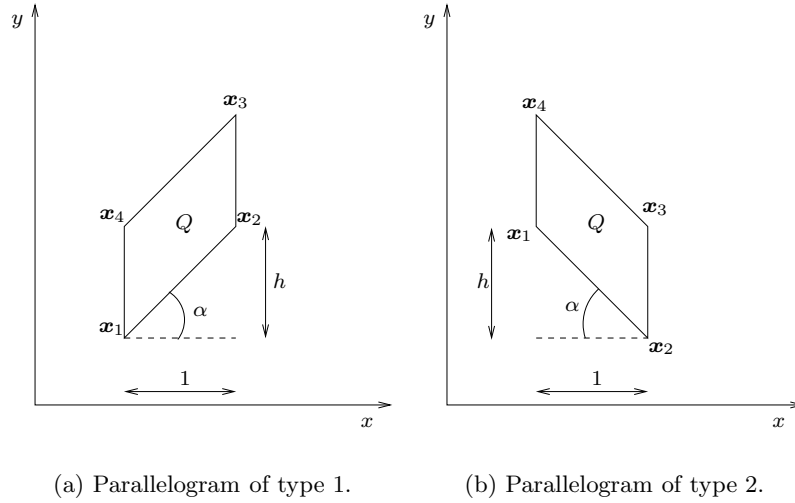


Figure 10.16: The angle α will be measure of the skewness of the parallelogram Q . The x -coordinates of the corners of Q is fixed.

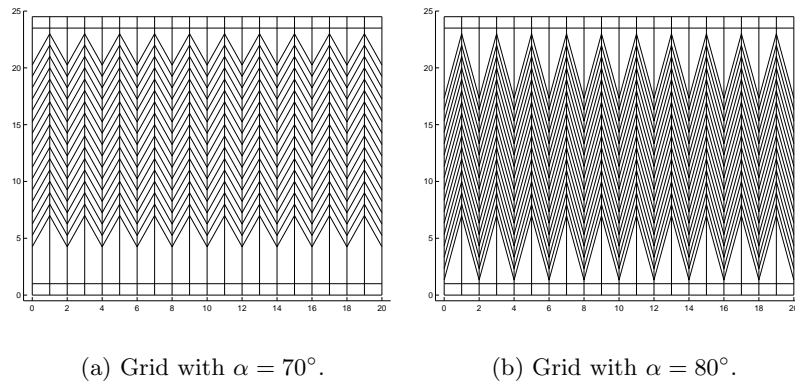


Figure 10.17: The type of grids that will be used in the testcase.

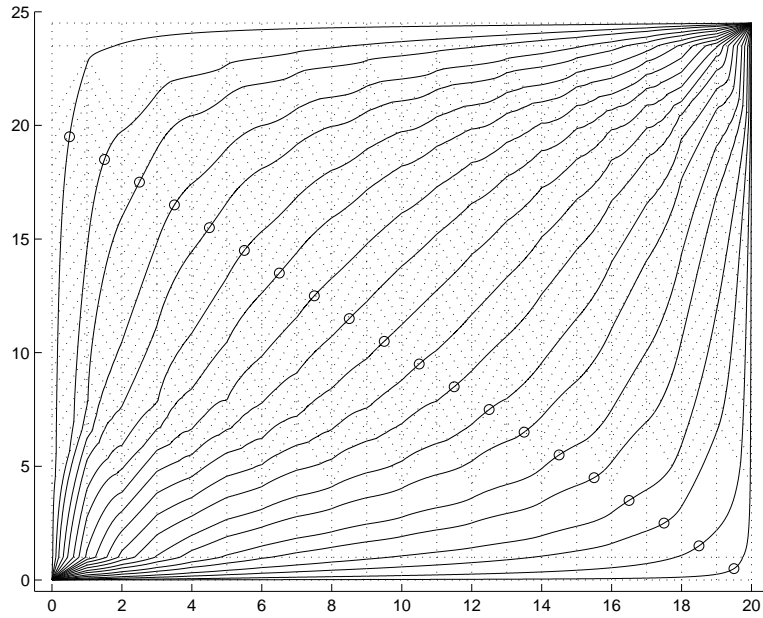
will end up and start at the same place for each grid. Both grids consists of same number of cells, and the boundary is the same.

The results are shown in Figure 10.18. For both figures the circle points are fixed locations in the domain, and each circle is the starting point for a streamline. Streamline tracing is done by starting the streamlines in the circle points and then tracing forward and backward until a source or a sink is reached.

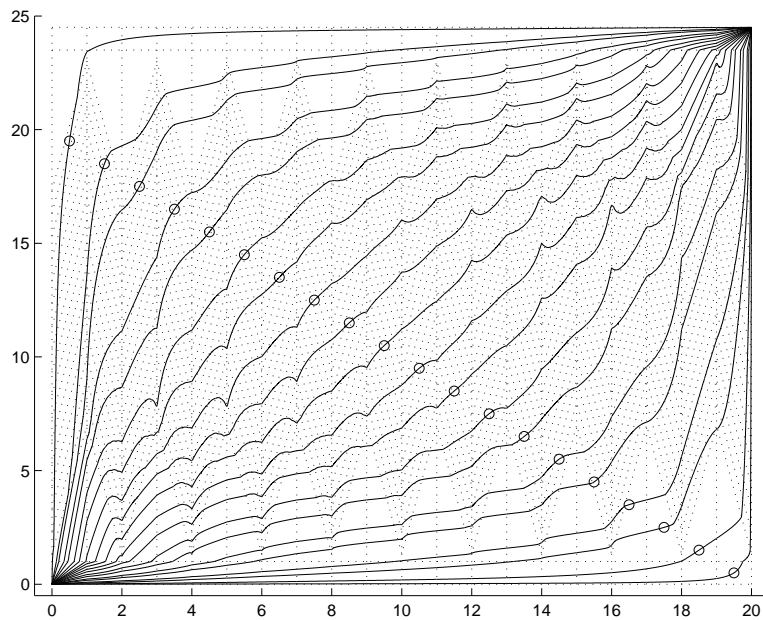
In Figure 10.19 the streamlines are plotted in the same grid for comparison. We observe that the streamlines are identical in a small neighbourhood of the circles, then the streamlines for $\alpha = 80^\circ$ starts to behave odd.

We now zoom in the picture between $8.3 < x < 9.3$ and $9 < y < 14$. See Figure 10.20. The dotted lines in the figure corresponds to the grid with $\alpha = 70^\circ$, whereas the dashed lines corresponds to the grid with $\alpha = 80^\circ$. Observe that the starting point of the streamline lies in different cells in the two grids.

If we will claim that the shape of the streamlines for $\alpha = 80^\circ$ is due to MPFA we have to know the analytical or true solution. For this problem we can find an analytical solution, and this will be investigated further in the future.



(a) Streamlines for grid with $\alpha = 70^\circ$.



(b) Streamlines for grid with $\alpha = 80^\circ$.

Figure 10.18: Streamlines for the grids.

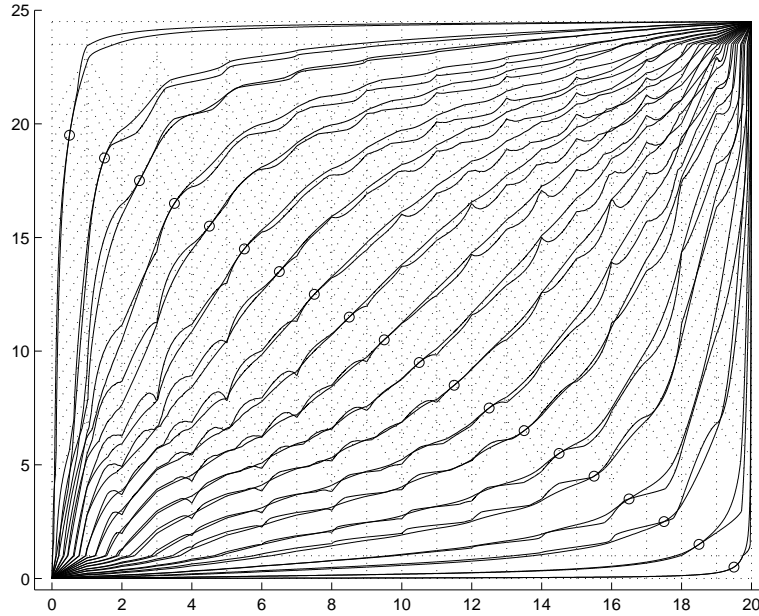


Figure 10.19: Streamlines for both grids plotted in the grid with $\alpha = 80^\circ$.

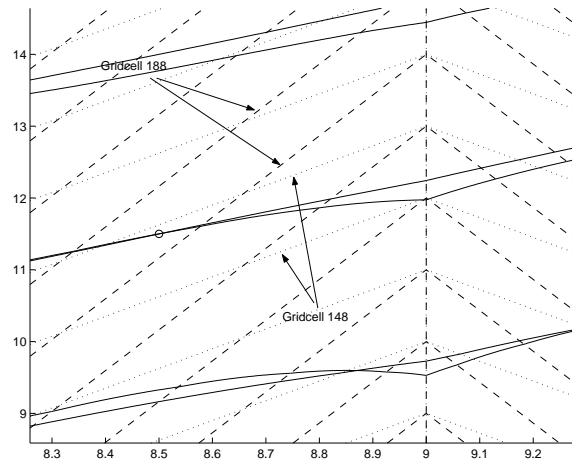


Figure 10.20: The streamlines for both grids zoomed in. Both grids are plotted. Dashed lines correspond to grid 2 and dotted lines correspond to grid 1. We see that the streamlines for grid 2 have cusps.

Chapter 11

Summary and Conclusions

11.1 Conclusion

In this thesis we have investigated the Prévost-method for streamline tracing. The method appears to be stable and accurate with respect to the shape of the streamlines, at least when the curvature of the streamlines is small. We have showed analytically that the method should preserve uniform flow, and this was also confirmed in the test cases.

Next we studied a circular velocity field. This example showed that the shape of the streamlines can be deformed when the direction of the velocity field is rapidly varying, as for the innermost circles in that example.

Introducing a velocity with divergence did not affect the quality of the streamlines, although we had no results to rely on for this case.

We then tested if the method was sensitive to perturbation of the grid. The result showed that the method was stable for the chosen testcase.

In the next section we verified that the Prévost method can give incorrect time-of-flight, and at last we studied the MPFA-algorithm using streamlines. This last example showed that streamlines can be used in certain cases to validate a flux or velocity distribution supplied by e.g. a numerical method.

We have shown that the modified Prévost-method and the Prévost-method give the same shape of the streamlines. Using a modified Prévost-method we can approximate the Jacobi determinant differently to obtain a better time-of-flight.

The errors in the shape of streamlines are due to (i) Inexact evaluation of fluxes on celledges. This is an error in the *input* to the method, and cannot be reduced by the method itself. It is commonly a numerical error in the solution of the flow equations. (ii) Interpolation of the velocity field in reference space. Different interpolation schemes can result in significantly different velocities, streamlines, and time-of-flight, as noted in [11].

The errors in the interpolation in the Prévost-method are due to the

flux-mapping from \mathcal{P} to \mathcal{R} and the linear interpolation within the cell in \mathcal{R} . There is no error associated with the bilinear transformation.

11.2 Further work

The results in this thesis shows that streamline tracing on irregular grids can be performed successfully. Developing a rigorous methodology for streamline tracing on irregular grids would be important step in estimating the accuracy of the solutions calculated by the streamline method. Investigating the errors introduced by the tracing method for 3D irregular grids will be one field of research.

Correct computation of time-of-flight is important both for particle tracking schemes and for the streamline method. A possible approach to improve TOF could be to approximate the Jacobi-determinant by the midpoint of the streamline within the cell.

Furthermore, the tracing algorithm should be implemented for two or three phases within the framework of the streamline method.

For the 2D tracing described in this thesis, there are a lot to investigate further, as noted in the results section. First we would like to try out a modified Prévost-method that evaluate J at the midpoint of a streamline, and compare the TOF calculations with the original method and with given analytical solutions.

We will investigate further how the method behaves on rapidly varying velocity fields. An error estimate relative to the velocity field would be of great help.

The derivation of a consistent tracing velocity using a direct transformation as described in Section 8.8, needs to be investigated further and extended to multiphase flow.

In the solution process the MPFA-method constructs interaction regions within each cell of the grid. In 2D there are four such subcells within each cell. At some of the edges of these subcells the flux is known. A possible strategy for improving the quality of the streamline when using the MPFA-method is to take advantage of this extra information.

Additionally, there is lot of numerical issues that has to be further investigated for the C++ implementation of the tracing algorithm.

Appendix A

Streamline Simulation

The streamline method used in streamline simulation involves two components; (i) tracing the streamline paths, and (ii) mapping 1D solutions along the streamlines.

A.1 The Governing IMPES Equations

The governing equations was derived in Section 3.1.

Equations (3.7) and (3.13) form the governing set of nonlinear equations for the IMPES method to be used in a streamline simulator. They are nonlinear since coefficients in each equation are dependent on the unknown variables (P or S_j). Although the equations are closely coupled together, they are different in mathematical behavior and as such can be solved differently. This is the underlying idea of the streamline IMPES method. Unlike a conventional finite-difference IMPES method, the use of streamlines allows one to transform Equation (3.13) into a pseudo 1D equation.

A.2 Coordinate Transformation Along Streamlines

In a conventional IMPES finite-difference simulator (3.13) is solved in its full three-dimensional form. With the streamline method (3.13) is decoupled into multiple 1D equations that are solved along streamlines. Solving multiple 1D equations along streamlines is much faster and more accurate than solving the full 3D problem.

To use the method (7.2) is first rewritten as

$$\frac{\partial \tau}{\partial s} = \frac{\phi}{|u_t|} \tag{A.1}$$

which can be further rewritten as

$$|u_t| \frac{\partial}{\partial s} \equiv \mathbf{u}_t \cdot \nabla = \phi \frac{\partial}{\partial \tau} \tag{A.2}$$

Substituting (A.2) in (3.13) gives

$$\frac{\partial S_j}{\partial t} + \frac{\partial f_j}{\partial \tau} + \frac{1}{\phi} \nabla \cdot \tilde{\mathbf{G}}_j = \frac{q_s f_{j,s}}{\phi} \quad (\text{A.3})$$

This is the governing pseudo 1D phase material balance equation transformed along a streamline coordinate. It is pseudo 1D since the gravity term is typically not aligned along the direction of a streamline.

Appendix B

Pollock tracing in 2D

In this Appendix we will give a more detailed exposition of Pollock tracing. The description here will be sufficient and necessary for an implementation in a streamline tracer for irregular grids. See Appendix C for an example.

Consider the case of streamline tracing for a unit square cell. See Figure B.1. It is sufficient to discuss this case only in order to trace streamlines

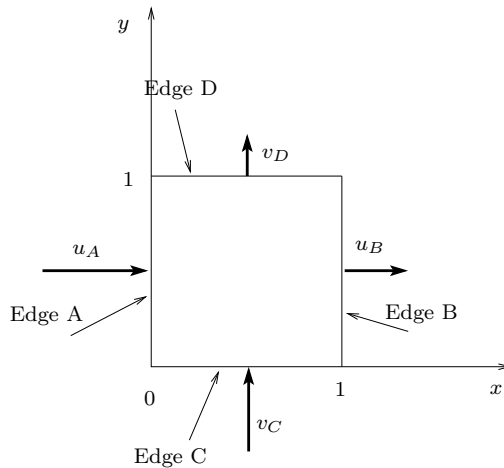


Figure B.1: Unit 2D cell with boundary velocities.

on any convex quadrilateral in 2D [24]. Assuming a steady velocity field, the edge velocities u_A, u_B, v_C, v_D will not change with time, and streamline tracing is equivalent to particle tracking. For a given particle starting at $P_0 = (x(0), y(0))$ at $t = 0$ within the cell in Figure B.1, we define the position to be given by

$$\mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j} \quad (\text{B.1})$$

Following Pollock [22], we assume a Raviart-Thomas [25] velocity field: within each spatial cell, the x -component is continuous piecewise-linear in the x -direction and discontinuous piecewise-constant in the y -direction, and

the y -component is the reverse. Thus we define the particle velocity within the cell by

$$\begin{aligned} \mathbf{v}(t) &= \frac{d\mathbf{r}}{dt} = \frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} \\ &= u(x(t), y(t))\mathbf{i} + v(x(t), y(t))\mathbf{j} = u(x(t))\mathbf{i} + v(y(t))\mathbf{j} \quad (\text{B.2}) \end{aligned}$$

Consider now the case in Figure B.2: we want to trace the path of the particle backwards in time, i.e., we want to calculate the particle's entry point when we are given the current position of the particle within the cell together with the velocities at the cell boundary.

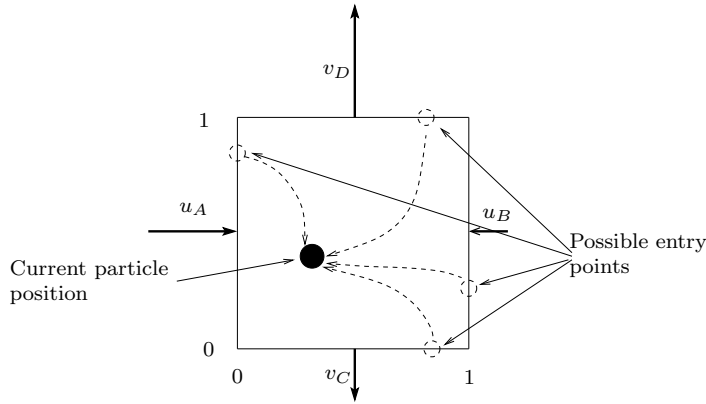


Figure B.2: Unit 2D cell with boundary velocities. We want to calculate the particle's entry point given it's current position.

Discussing this case will be sufficient to trace any streamline from a given point forward and backward in time. Suppose we want to trace a streamline forward in time from a given point. Then this is equivalent to trace the same streamline backwards in time with the cell edge velocities reversed. This can be seen from the following argument: The velocity vector (B.2) is everywhere tangent to the streamline, when reversing the cell edge velocities this vector will also be reversed, i.e., $\mathbf{v}_r(x, y) = -\mathbf{v}(x, y)$. Here $\mathbf{v}_r(x, y)$ is the reversed velocity vector. Hence the two streamlines have the same tangent vectors everywhere. Since they pass through the same point they must be the same. Note also that $|\mathbf{v}| = |\mathbf{v}_r|$ so the speed is the same at any point. Then the travel time must also be the same.

B.1 Find the travel time and the entry edge

Consider first $x(t)$, the x -coordinate of the particle's position. Using linear interpolation of $u(t)$, we must have

$$\frac{dx}{dt} = u(t) = G_x x(t) + u_A \quad (\text{B.3})$$

where $G_x = u_B - u_A$ is the velocity gradient. By the steady-state assumption G_x , u_A and u_B is constant.

This gives us an ordinary differential equation for $x(t)$,

$$\begin{cases} \frac{dx}{G_x x + u_A} = dt, & G_x \neq 0 & \text{(B.4)} \\ \frac{dx}{u_A} = dt, & G_x = 0, \quad u_A \neq 0 & \text{(B.5)} \\ \frac{dx}{dt} = 0, & G_x = 0, \quad u_A = 0 & \text{(B.6)} \end{cases}$$

We are given a point, P , within the cell, which we will denote by $(x(T), y(T))$, where T is the time used for a particle starting at a point P_0 on a cell edge or from a source within the cell to reach P . Note that T itself is not known yet. This notation might be a bit confusing as usually the known point is denoted by P_0 . We will still refer to P_0 as the starting point, but keep in mind that it is not known yet. We refer to P as the end point, and T will be the travel time.

Next we want to integrate the previous equation from $t = 0$ to $t = T$. Note that Equation (B.4) can only be integrated if the denominator of the integrand is not zero over the integration interval. This happens if the integrand evaluated at the left limit of the integration interval has opposite sign of the integrand evaluated at the right limit of the integration interval. Physically this corresponds to that a particle moving in one dimension in a stationary velocity field can never advance from a point with a positive velocity to a point with a negative velocity or vice versa in a finite time. Note from (B.3) that $u(T) = G_x x(T) + u_A$ and $u(0) = G_x x(0) + u_A$. Now integration yields

$$T = \begin{cases} \text{Undefined}, & G_x \neq 0, \quad u(0) = 0 \\ \text{Undefined}, & G_x \neq 0, \quad u(T)/u(0) \leq 0 \\ \frac{1}{G_x} \ln \left| \frac{G_x x(T) + u_A}{G_x x(0) + u_A} \right| = \frac{1}{G_x} \ln \frac{u(T)}{u(0)}, & G_x \neq 0, \quad u(T)/u(0) > 0 \\ \frac{x(T) - x(0)}{u_A}, & G_x = 0, \quad u_A \neq 0 \\ T, x(T) = x(0), & G_x = 0, \quad u_A = 0 \end{cases} \quad \text{(B.7)}$$

Next we want to exclude negative times. We do this by setting $T = \infty$ if $T < 0$. We also set $T = \infty$ if T is undefined according to the previous

discussion. Let's first summarize the conditions giving $T = \infty$,

$$\text{Cx1: } G_x \neq 0, \quad u(0) = 0 \tag{B.8}$$

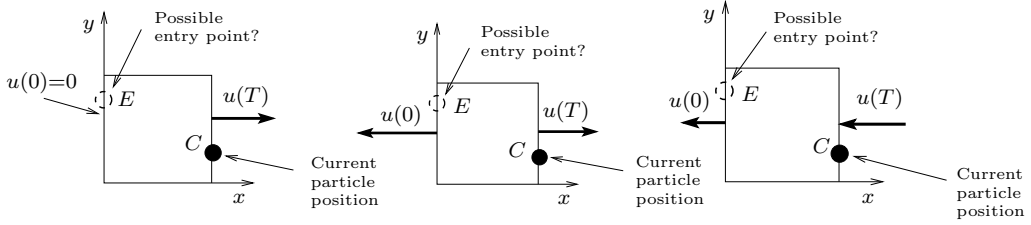
$$\text{Cx2: } G_x \neq 0, \quad \frac{u(T)}{u(0)} \leq 0 \tag{B.9}$$

$$\text{Cx3: } G_x \neq 0, \quad \frac{u(T)}{u(0)} > 0, \quad \frac{1}{G_x} \ln \frac{u(T)}{u(0)} < 0 \tag{B.10}$$

$$\text{Cx4: } G_x = 0, \quad u_A \neq 0, \quad \frac{x(T) - x(0)}{u_A} < 0 \tag{B.11}$$

$$\text{Cx5: } G_x = 0, \quad u_A = 0 \tag{B.12}$$

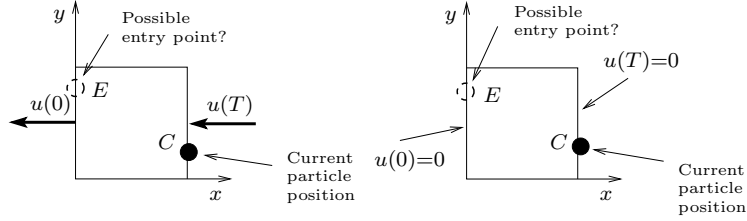
See Figure B.3 for some examples of these.



(a) Cx1: Since the particle at E has speed 0, it will never reach C .

(b) Cx2: Since the particle at E moves to the left, whereas the particle at C moves to the right, E cannot be an entry point.

(c) Cx3: The particle at C will reach E at a later time, thus E cannot be an entry point.



(d) Cx4: Same as Cx3, except that $u(0)=u(T)$.

(e) Cx5: Since the particle at C doesn't move it cannot enter at E .

Figure B.3: Negative times. We have set $x(0) = 0$ and $x(T) = 1$. These cases are not physically possible, so we indicate that by setting $T = \infty$.

Set

$$\text{Cx} = \text{Cx1 or Cx2 or Cx3 or Cx4 or Cx5} \tag{B.13}$$

Then we have

$$T = \begin{cases} \infty, & \text{If } Cx \\ \frac{1}{G_x} \ln \frac{u(T)}{u(0)}, & G_x \neq 0, \quad \frac{1}{G_x} \ln \frac{u(T)}{u(0)} \geq 0 \\ \frac{x(T)-x(0)}{u_A}, & G_x = 0, \quad u_A \neq 0, \quad \frac{x(T)-x(0)}{u_A} \geq 0 \end{cases} \quad (\text{B.14})$$

For the y -direction we assume that the velocity varies linearly with the y -coordinate:

$$\frac{dy}{dt} = v(t) = G_y y(t) + v_C \quad (\text{B.15})$$

where $G_y = v_D - v_C$ is the velocity gradient in the y -direction. A similar calculation as above for the y -direction gives

$$T = \begin{cases} \infty, & \text{If } Cy \\ \frac{1}{G_y} \ln \frac{v(T)}{v(0)}, & G_y \neq 0, \quad \frac{1}{G_y} \ln \frac{v(T)}{v(0)} \geq 0 \\ \frac{y(T)-y(0)}{v_C}, & G_y = 0, \quad v_C \neq 0, \quad \frac{y(T)-y(0)}{v_C} \geq 0 \end{cases} \quad (\text{B.16})$$

Now the particle can enter the cell through the faces A, B, C, D or the particle never enters the cell (see Figure B.4 for an example).

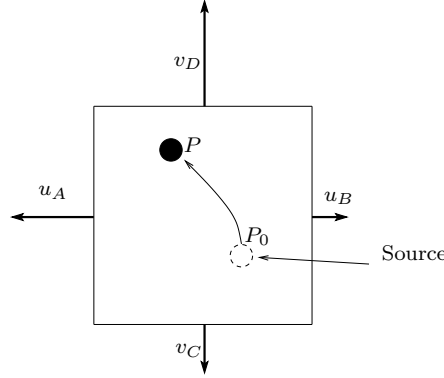
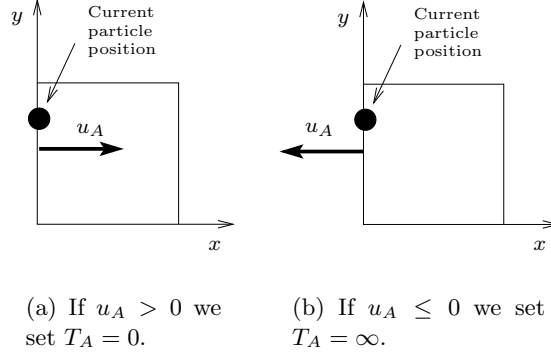


Figure B.4: Unit 2D cell with boundary velocities. A special case where the particle never enters the cell rather it starts at a point inside the cell due to a source in the cell.

We can find the correct entry face (if the particle enters the cell) by the following procedure. Consider first the time, T_A , for the particle to reach the point P starting from the line $x=0$ (this line contains edge A) at $t=0$. Then setting $x(0)=0$ implies by (B.3) that $u(0)=u_A$. Inserting this in Equation (B.14) we can find T_A , but consider first the special case $x(T) = 0$. In this case $u(T)=u(0)$ and $u(T)/u(0)=1$, and from (B.14) we get $T_A = (1/G_x) \ln 1 = 0$. If $u_A > 0$ this is correct because in this case the end point and starting point will be the same. If $u_A \leq 0$ then we set $T_A = \infty$ because in this case we will not consider face A as an entry face, as

Figure B.5: The special case for $x(0) = x(T) = 0$.

the particle certainly will enter the cell through an other face (if $u_A \neq 0$). See Figure B.5.

Accordingly we define the additional condition

$$\text{CxA6: } x(0) = 0, \quad x(T) = 0, \quad u_A \leq 0 \quad (\text{B.17})$$

We define CxA1, CxA2, CxA3, CxA4 and CxA5 by setting

$$T = T_A, \quad u(0) = u_A, \quad x(0) = 0 \quad (\text{B.18})$$

in (B.8), (B.9), (B.10), (B.11) and (B.12). And define

$$\text{CxA} = \text{CxA1 or CxA2 or CxA3 or CxA4 or CxA5 or CxA6} \quad (\text{B.19})$$

Then

$$T_A = \begin{cases} \infty, & \text{If CxA} \\ \frac{1}{G_x} \ln \frac{u(T_A)}{u_A}, & G_x \neq 0, \frac{1}{G_x} \ln \frac{u(T_A)}{u_A} > 0 \\ \frac{x(T_A)}{u_A}, & G_x = 0, \frac{x(T_A)}{u_A} > 0 \end{cases} \quad (\text{B.20})$$

Next calculate the time for the particle to reach P starting from the line $x = 1$, containing edge B . Setting $T = T_B$, $u(0) = u_B$ and $x(0) = 1$ in (B.14) gives us

$$T_B = \begin{cases} \infty, & \text{If CxB} \\ \frac{1}{G_x} \ln \frac{u(T_B)}{u_B}, & G_x \neq 0, \frac{1}{G_x} \ln \frac{u(T_B)}{u_B} > 0 \\ \frac{x(T_B)-1}{u_A}, & G_x = 0, \frac{x(T_B)-1}{u_A} > 0 \end{cases} \quad (\text{B.21})$$

where CxB is defined similar to CxA and handles the special case $x(T) = 1$.

$$T_C = \begin{cases} \infty, & \text{If CyC} \\ \frac{1}{G_y} \ln \frac{v(T_C)}{v_C}, & G_y \neq 0, \frac{1}{G_y} \ln \frac{v(T_C)}{v_C} > 0 \\ \frac{y(T_C)}{v_C}, & G_y = 0, \frac{y(T_C)}{v_C} > 0 \end{cases} \quad (\text{B.22})$$

corresponding to the line $y = 0$, and

$$T_D = \begin{cases} \infty, & \text{If CyD} \\ \frac{1}{G_y} \ln \frac{v(T_D)}{v_D}, & G_y \neq 0, \frac{1}{G_y} \ln \frac{v(T_D)}{v_D} > 0 \\ \frac{y(T_D)-1}{v_C}, & G_y = 0, \frac{y(T_D)-1}{v_C} > 0 \end{cases} \quad (\text{B.23})$$

for the line $y = 1$.

Now the minimum of T_A, T_B, T_C and T_D will give the particle's travel time, T ,

$$T = \min(T_A, T_B, T_C, T_D) \quad (\text{B.24})$$

This is the time for the particle to reach its current position starting from a point on the cell boundary (if the particle enters the cell).

B.2 Find the entry point and the trajectory

Consider first the case $T < \infty$. We will now calculate the particle's entry point as it is needed in the equation for the trajectory. The particle's entry point $(x(0), y(0))$ is given by (B.14) and (B.16) by substituting $u(0) = G_x x(0) + u_A$ and $v(0) = G_y y(0) + v_C$ we get

$$T = \begin{cases} \frac{1}{G_x} \ln \left[\frac{u(T)}{G_x x(0) + u_A} \right], & G_x \neq 0 \\ \frac{x(T) - x(0)}{u_A}, & G_x = 0 \end{cases} \quad (\text{B.25})$$

$$T = \begin{cases} \frac{1}{G_y} \ln \left[\frac{v(T)}{G_y y(0) + v_C} \right], & G_y \neq 0 \\ \frac{y(T) - y(0)}{v_C}, & G_y = 0 \end{cases} \quad (\text{B.26})$$

solving these equations for $x(0)$ and $y(0)$ we get

$$x(0) = \begin{cases} \frac{1}{G_x} (e^{-G_x T} u(T) - u_A), & G_x \neq 0 \\ x(T) - u_A T, & G_x = 0 \end{cases} \quad (\text{B.27})$$

$$y(0) = \begin{cases} \frac{1}{G_y} (e^{-G_y T} v(T) - v_C), & G_y \neq 0 \\ y(T) - v_C T, & G_y = 0 \end{cases} \quad (\text{B.28})$$

The particle's trajectory within the cell is now given by (B.14) and (B.16). Setting $T = t$ in those equations we get

$$x(t) = \begin{cases} \frac{1}{G_x} ((G_x x(0) + u_A) e^{G_x t} - u_A), & G_x \neq 0, & 0 \leq t \leq T \\ u_A t + x(0), & G_x = 0, & 0 \leq t \leq T \end{cases} \quad (\text{B.29})$$

$$y(t) = \begin{cases} \frac{1}{G_y} ((G_y y(0) + v_C) e^{G_y t} - v_C), & G_y \neq 0, & 0 \leq t \leq T \\ v_C t + y(0), & G_y = 0, & 0 \leq t \leq T \end{cases} \quad (\text{B.30})$$

B.3 Forward tracing

Consider for a moment tracing a streamline forward from a given point P_0 within the cell. See Figure B.6(a). As described in the beginning of this

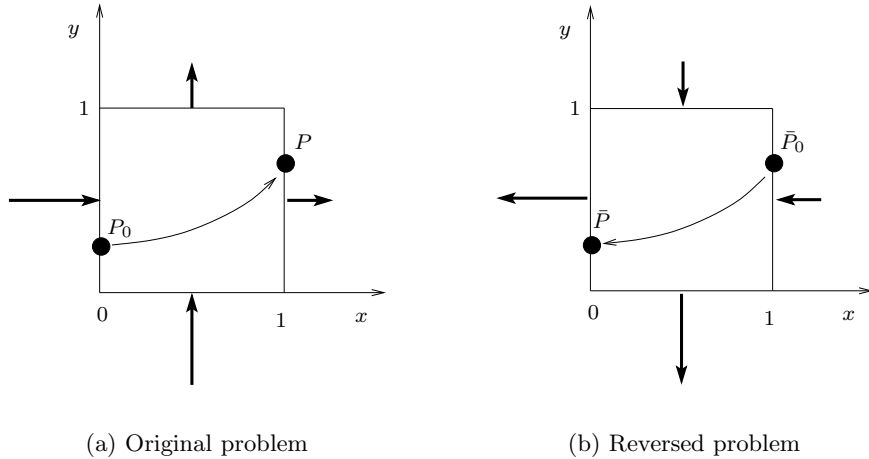


Figure B.6: Tracing forward.

section this introduces no major problem. We will only comment on some notational issues here. The forward tracing is done by reversing the cell edge velocities and then carrying out backward tracing. See figure B.6(b). For the reversed problem we will have that $\bar{P} = P_0$ and $\bar{P}_0 = P$ as in the figure (barred variables will refer to the reversed problem). This is consistent with the backward tracing algorithm described earlier in this section, which requires P to be the known point. Then this algorithm will find \bar{P}_0 . The last step should then be to set $P = \bar{P}_0$ to get the end point for the forward tracing algorithm.

The trajectory when $T < \infty$ is still given by (B.29) and (B.30).

B.4 The special case of a source or a sink

If $T < \infty$ the particle will reach the current point starting from a point at the boundary in a finite time and it cannot come from a source within the cell (since then it would be another particle). The further movement of the particle is yet unknown. The particle could end up in sink or not. If we know that the current point is also on the boundary, then there is no sink inside the cell.

We will consider another viewpoint, and assume that $T = \infty$. What can we then say about the particle's trajectory? When $T = \infty$ we must have from (B.24) that all of T_A , T_B , T_C and T_D is infinite. We could then

investigate when it is possible that all of CxA, CxB, CxC, CxD, could occur at the same time, to give some condition for an infinite travel time. We will instead use a simpler argument. One trivial condition for an infinite travel time is that $u_A = u_B = v_C = v_D = 0$. Then there is no movement at all inside the cell. And we will not say that there is a source or sink in the cell. Assume that this isn't the case. Let's define

$$u_0\text{-line: } \{x = x_0, 0 \leq y \leq 1 : u(x_0) = 0\} \quad (\text{B.31})$$

$$v_0\text{-line: } \{y = y_0, 0 \leq x \leq 1 : v(y_0) = 0\} \quad (\text{B.32})$$

If $T = \infty$ there must exist both an u_0 -line and a v_0 -line within the cell. Because, suppose there is no u_0 -line then the particle will at any point within the cell have an everywhere positive or everywhere negative x-component of velocity. So the particle has to reach the current point within a finite time. The same argument goes for the y -direction.

Suppose then there is both an u_0 -line and a v_0 -line within the cell. If the velocity of the particle at the current point is not zero, it must come from a source within the cell. On the other hand, if the velocity is currently zero, then this is a sink.

To sum up the above discussion we have for the x -direction, if $T = \infty$, either of the following

1. $u_A = u_B = v_C = v_D = 0$

In this case we have that

$$x(0) = x(T) = x(t), \quad t \geq 0 \quad (\text{B.33})$$

$$y(0) = y(T) = y(t), \quad t \geq 0 \quad (\text{B.34})$$

This case is not very interesting.

2. $u_A * u_B \leq 0$ **and** $v_C = v_D = 0$:

Then $x(0)$ is the stagnation point that gives zero velocity. That is $x(0)$ satisfies $G_x x(0) + u_A = 0$. This gives

$$x(0) = -\frac{u_A}{G_x} \quad (\text{B.35})$$

$$y(0) = y(T) = y(t), \quad t \geq 0 \quad (\text{B.36})$$

3. $v_C * v_D \leq 0$ **and** $u_A = u_B = 0$:

Then $y(0)$ is the stagnation point that gives zero velocity. So $y(0)$ satisfies $G_y y(0) + v_C = 0$. This gives

$$x(0) = x(T) = x(t), \quad t \geq 0 \quad (\text{B.37})$$

$$y(0) = -\frac{v_C}{G_y} \quad (\text{B.38})$$

4. $u_A * u_B \leq 0$ and $v_C * v_D \leq 0$

We then have

$$x(0) = -\frac{u_A}{G_x} \quad (\text{B.39})$$

$$y(0) = -\frac{v_C}{G_y} \quad (\text{B.40})$$

The last step is to calculate the trajectory when the travel time is infinite for case 2,3 and 4 above. Consider first forward tracing. We must have a sink within the cell (at least for case 4). See Figure B.7(a). We reverse

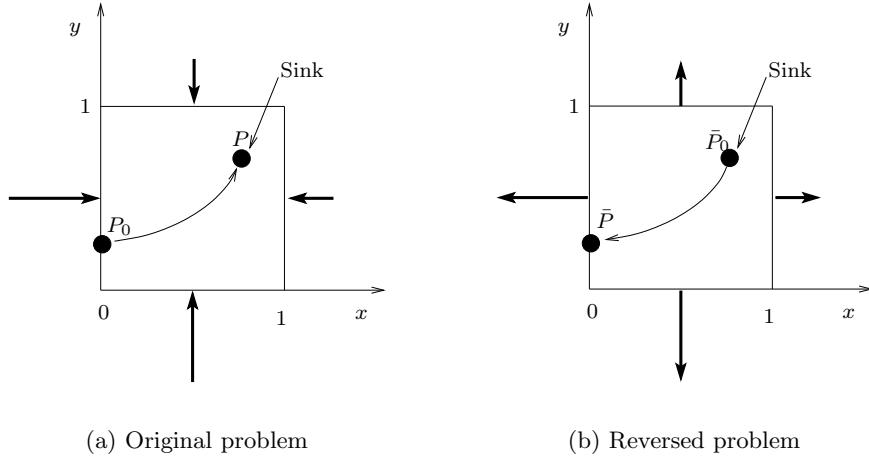


Figure B.7: A sink within the cell.

the velocities and trace backwards. See Figure B.7(b). We then find for case 2 and 4, by (B.35) and (B.39), that $\bar{x}(0) = -\frac{\bar{u}_A}{G_x} = -\frac{u_A}{G_x}$. We then set $x(T) = \bar{x}(0)$. Now consider using equations (B.29) and (B.30) for the trajectory. For the case of a sink within the cell we must have

$$G_x < 0, \quad G_y < 0 \quad (\text{B.41})$$

see Figure B.7(a). Then using (B.29), we have

$$\lim_{t \rightarrow \infty} x(t) = \lim_{t \rightarrow \infty} \frac{1}{G_x} ((G_x x(0) + u_A)e^{G_x t} - u_A) = -\frac{u_A}{G_x} \quad (\text{B.42})$$

since $G_x < 0$. For the y -direction in case 3 and 4, we also have using (B.30)

$$\lim_{t \rightarrow \infty} y(t) = -\frac{v_C}{G_y} \quad (\text{B.43})$$

This shows that we could use (B.29) and (B.30) for the trajectory in this case. Since for times greater than a certain value, say $t = 1000$ the points

will lie so close to P that they can't be separated from P by eye. So for the case of plotting the streamline we only have to consider the interval $t \in [0, T_{max}]$ where T_{max} is "not too big".

Next consider tracing backwards for cases 2,3 and 4 above, then there must be a source within the cell (at least for case 4). See Figure B.8. Now, since P_0 lies at the source point we have that $G_x x(0) + u_A = 0$, and

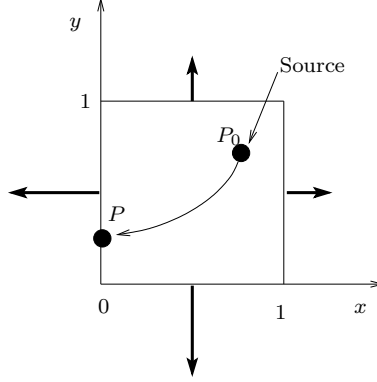


Figure B.8: A source within the cell.

$G_y y(0) + v_C = 0$. Equations (B.29) and (B.30) then gives for case 4,

$$x(t) = -\frac{u_A}{G_x}, \quad 0 < t < \infty \quad (\text{B.44})$$

$$y(t) = -\frac{v_C}{G_y}, \quad 0 < t < \infty \quad (\text{B.45})$$

So all points on the trajectory cluster at P_0 for all finite times. This representation cannot be used to plot the streamline. But we can fix the problem by reversing the celledge velocities and tracing the streamline forward instead. Setting $\bar{P}_0 = P$, and writing (B.29) and (B.30) as

$$\bar{x}(t) = \frac{1}{\bar{G}_x} \left((\bar{G}_x \bar{x}(0) + \bar{u}_A) e^{\bar{G}_x t} - \bar{u}_A \right) \quad (\text{B.46})$$

$$\bar{y}(t) = \frac{1}{\bar{G}_y} \left((\bar{G}_y \bar{y}(0) + \bar{v}_C) e^{\bar{G}_y t} - \bar{v}_C \right) \quad (\text{B.47})$$

where the barred variables refers to the reversed trajectory. We must have

$$\bar{G}_x = -G_x, \quad \bar{G}_y = -G_y \quad (\text{B.48})$$

$$\bar{u}_A = -u_A, \quad \bar{v}_C = -v_C \quad (\text{B.49})$$

$$\bar{x}(0) = x(T), \quad \bar{y}(0) = y(T) \quad (\text{B.50})$$

inserting i (B.48), (B.49) and (B.50) in (B.46) and (B.47) we get

$$\bar{x}(t) = \frac{1}{G_x} ((G_x \bar{x}(0) + u_A) e^{-G_x t} - u_A) \quad (\text{B.51})$$

$$\bar{y}(t) = \frac{1}{G_y} ((G_y \bar{y}(0) + v_C) e^{-G_y t} - v_C) \quad (\text{B.52})$$

Since $G_x > 0$ and $G_y > 0$ for a source cell this equation is of the same form as the equations for the sink cell above, and the same analysis applies.

Appendix C

Tracing on irregular grids

We will illustrate the method of streamline tracing for irregular grids for the case single-phase incompressible horizontal flow in 2D for a homogeneous medium. The solution domain will be 20x20 square aligned with the coordinate-axes with no flow boundaries. We discretize the domain with an irregular quadrilateral grid, and introduce a source and a sink of unit strength, see Figure C.1(a).

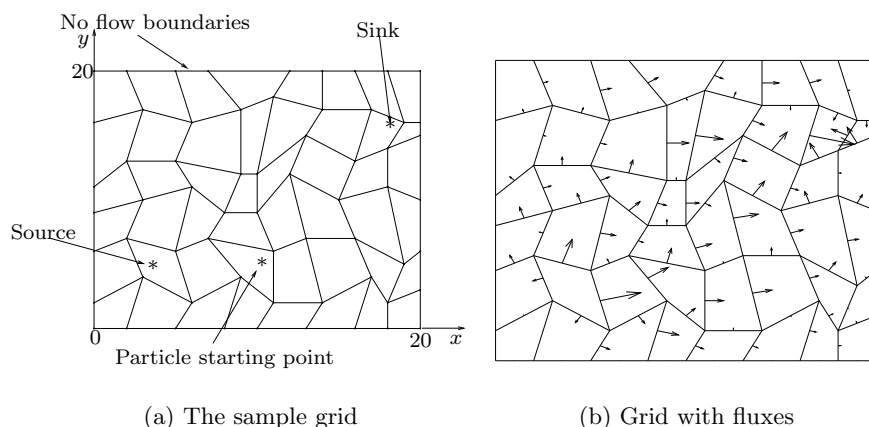


Figure C.1: Grid and edge fluxes

1. Solve the flow equations

The first step is to solve the flow equations for the given grid as discussed in section 3.2. This will give us celledge fluxes as shown in Figure C.1(b). The arrows on the Figure should not be confused with velocities, they are drawn to visualize the size of the edgefluxes.

2. Find the gridcell containing the point

Assume a starting point $P = (x, y)$ within the solution domain is given,

and we want to trace the streamline passing through that point. We must then find the cell, C , in the grid containing the point. This can be done using the algorithm described in Appendix D. We search through the grid cell by cell and use the algorithm on each cell.

3. Inverse map of point

The next step is to map the starting point within the cell C to \mathcal{R} using the inverse bilinear map as described in Appendix E. See Figure C.2.

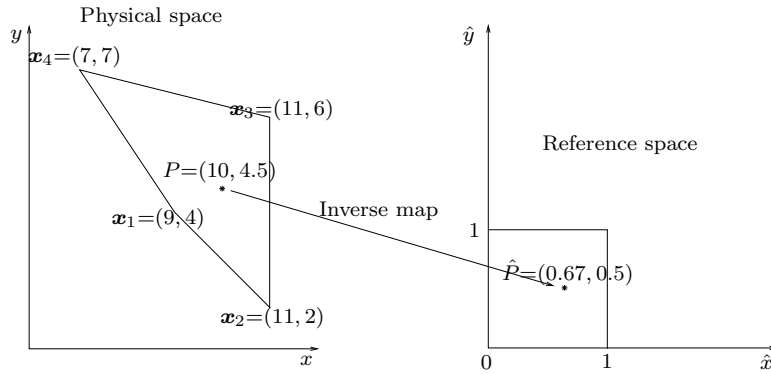


Figure C.2: The inverse map

4. Map of fluxes

Then we map the edge fluxes in \mathcal{P} to edge fluxes in \mathcal{R} . We will use the Prévost-method (section 8.7) and transform the fluxes according to (8.57) and (8.17). See Figure C.3.

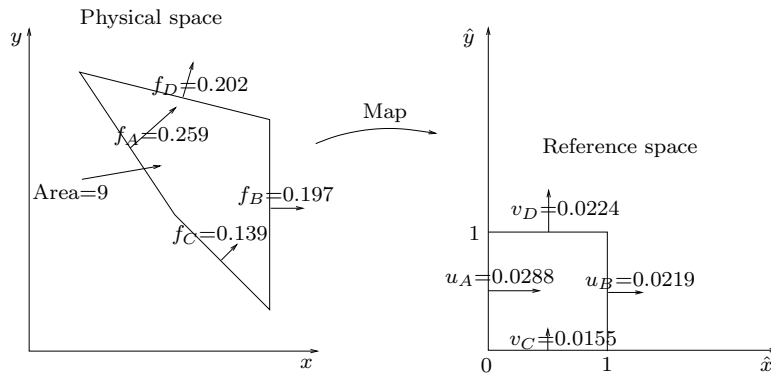


Figure C.3: Map of fluxes

5. Trace backwards

We want to trace the streamline starting at a point \hat{P}_0 on the celledge

to the known point \hat{P} within the cell. We first find the correct entry face. Set

$$\hat{P} = (\hat{x}(T), \hat{y}(T)) = (0.67, 0.5) \quad (\text{C.1})$$

$$G_x = u_B - u_A = -0.00696 \quad (\text{C.2})$$

$$G_y = v_D - v_C = 0.00696 \quad (\text{C.3})$$

$$u(T) = G_x \hat{x}(T) + u_A = 0.0241 \quad (\text{C.4})$$

$$v(T) = G_y \hat{y}(T) + v_C = 0.0190 \quad (\text{C.5})$$

then from (B.20) we have

$$T_A = 25.6 \quad (\text{C.6})$$

From (B.21) and CxB3 in that equation we have that

$$T_B = \infty \quad (\text{C.7})$$

From (B.22) we have

$$T_C = 29.3 \quad (\text{C.8})$$

and from (B.23) and CyD3 we have

$$T_D = \infty \quad (\text{C.9})$$

Then since

$$\min(T_A, T_B, T_C, T_D) = T_A \quad (\text{C.10})$$

the particle will exit face A , and consequently we know that $\hat{x}(0) = 0$. Set $T = T_A$. Then from (C.5) and (B.28) we find that

$$\hat{y}(0) = \frac{1}{G_y} (e^{-G_y T} v(T) - v_C) = 0.061 \quad (\text{C.11})$$

and the trajectory is given by (B.29) and (B.30),

$$\hat{x}(t) = \frac{1}{G_x} ((G_x \hat{x}(0) + u_A) e^{G_x t} - u_A), \quad 0 < t < T = 25.6 \quad (\text{C.12})$$

$$\hat{y}(t) = \frac{1}{G_y} ((G_y \hat{y}(0) + v_C) e^{G_y t} - v_C), \quad 0 < t < T = 25.6 \quad (\text{C.13})$$

See Figure C.4(a). We can now map this streamline back to the physical space using (6.1). See Figure C.4(b).

6. Trace forwards

Next, we trace the streamline forwards in the same cell, to obtain a complete trajectory within the first cell. As pointed out in Appendix

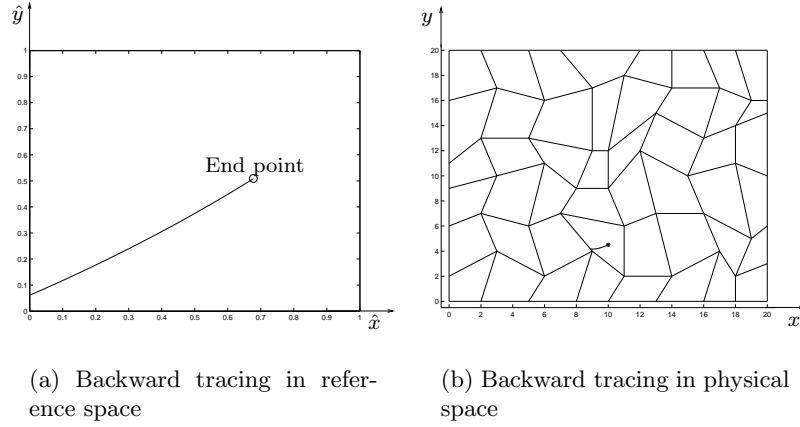


Figure C.4: Backward tracing

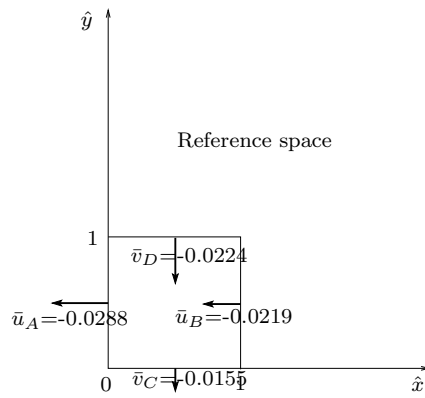


Figure C.5: Reversed velocities

B, tracing forwards is the same as tracing backwards with the celledge velocities reversed. So we set

$$\bar{u}_A = -u_A, \quad \bar{u}_B = -u_B, \quad \bar{v}_C = -v_C, \quad \bar{v}_D = -v_D \quad (\text{C.14})$$

As in Appendix B.3, barred variables will refer to the reversed problem. See Figure C.5. We should then trace backwards from $\hat{\hat{P}} = (0.67, 0.5)$, but it will be easier to use $\hat{\hat{P}} = \hat{P}_0$, where \hat{P}_0 was found in step 5. Although we traces the same curve twice it will be no difference in computation time since we are not plotting anything currently. We then use the same method as in step 5. We now have

$$\hat{\hat{P}} = (\hat{\hat{x}}(\bar{T}), \hat{\hat{y}}(\bar{T})) = (0, 0.061) \quad (\text{C.15})$$

$$\bar{G}_x = \bar{u}_B - \bar{u}_A = 0.00696 \quad (\text{C.16})$$

$$\bar{G}_y = \bar{v}_D - \bar{v}_C = -0.00696 \quad (\text{C.17})$$

$$\bar{u}(\bar{T}) = \bar{G}_x \hat{\hat{x}}(\bar{T}) + \bar{u}_A = \bar{u}_A = -0.0288 \quad (\text{C.18})$$

$$\bar{v}(\bar{T}) = \bar{G}_y \hat{\hat{y}}(\bar{T}) + \bar{v}_C = -0.0159 \quad (\text{C.19})$$

We then have by (B.20) and CxA6 that

$$\bar{T}_A = \infty \quad (\text{C.20})$$

From (B.21) we have

$$\bar{T}_B = 39.4 \quad (\text{C.21})$$

(B.22) and CyC3 gives

$$\bar{T}_C = \infty \quad (\text{C.22})$$

and from (B.23) we have

$$\bar{T}_D = 49.43 \quad (\text{C.23})$$

so

$$\min(\bar{T}_A, \bar{T}_B, \bar{T}_C, \bar{T}_D) = \bar{T}_B \quad (\text{C.24})$$

The particle will then enter face B , and consequently we know that $\hat{\hat{x}}(0) = 1$. Set $\bar{T} = \bar{T}_B$, then from (C.19) and (B.28) we find that

$$\hat{\hat{y}}(0) = \frac{1}{\bar{G}_y} \left(e^{-\bar{G}_y \bar{T}} \bar{v}(\bar{T}) - \bar{v}_C \right) = 0.780 \quad (\text{C.25})$$

We then set $\hat{\hat{P}} = \hat{P}_0$ and $T = \bar{T}$. The trajectory is then found in a similar fashion as for step 5. See Appendix B.3. The results are plotted in Figure C.6.

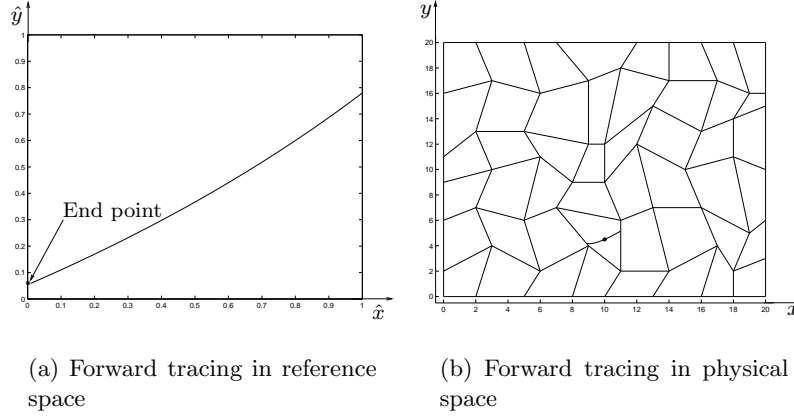


Figure C.6: Forward tracing

7. Make `CellData` object

Actually we don't plot any streamlines at the current stage, rather we save the information necessary to reconstruct it at a later time. Observe that the streamline is completely determined by the parameters

$$u_A, u_B, v_C, v_D, P_0, P, T, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \quad (\text{C.26})$$

The parameter P will be needed for plotting streamlines in cells containing sources as described in Appendix B.4. Given these parameters the trajectory in the reference space is given by (B.29) and (B.30), and the points on this trajectory can be mapped using (6.1) to obtain a streamline in the physical space within the current cell. To implement this in a *C++* program, we save the parameters (C.26) in a *CellData* object, that will be included in a *SLine* object which is a *STL List* containing all the *CellData* objects for a complete streamline.

8. Find the previous cell on the streamline

In order to find the previous cell in the grid. We must know how the grid is structured. In this work we will be using the grid described in Section 5.1. So define

$$n = \text{Current gridcell number} \quad (\text{C.27})$$

$$n_x = \text{Number of gridcells in the } x\text{-direction} \quad (\text{C.28})$$

$$n_y = \text{Number of gridcells in the } y\text{-direction} \quad (\text{C.29})$$

For this sample case we have that $n=12$, $n_x=8$, $n_y=6$. See figure C.7.

A travel time of $T = \infty$ in the backward tracing part of the algorithm means that this cell is a source. So in that case this cell is the starting cell of a streamline, and there will be no previous cell.

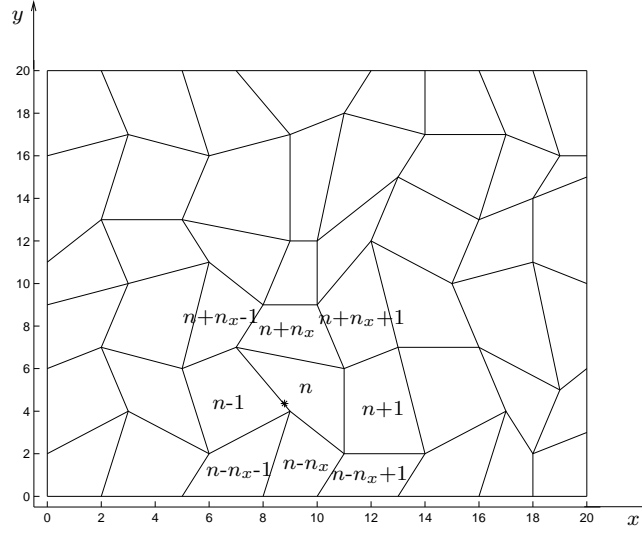


Figure C.7: Find the previous cell

Now, assume that $T < \infty$, and that a point $\hat{P}_0 = (\hat{x}(0), \hat{y}(0))$ on the cell boundary in the reference space is given corresponding to a point P_0 in the physical space. We must here assure that if the point \hat{P}_0 lies for instance on the left boundary, then $\hat{x}(0)$ should be exactly equal to zero, not approximately due to round off errors. This will be easy to accomplish, since by (B.24) we know that if, e.g., $T = T_A$ then $\hat{x}(0)$ should be exactly equal to zero, and we do not use Equation (B.27), which probably would give an approximate value.

So consider first the case $\hat{x}(0) = 0$ and the case that the physical cell is one of the left boundary cells of the solution domain. This is true if $n \% n_x = 0$, where $\%$ denotes the modulus operator. If this is the case, then the streamline ends with this cell. This could not happen in this sample case, since we have no-flow boundaries, but it might happen in other cases.

So assume we are not on the left boundary. Then the previous physical cell can be one of $n-1$, $n-n_x-1$, $n+n_x-1$, depending on the value of $\hat{y}(0)$. We will not inspect the value of $\hat{y}(0)$ at this point though, instead we introduce a variable Δ_x , and set $\Delta_x = -1$.

Now, suppose $\hat{x}(0) = 1$, then if the physical cell is not a right boundary cell of the solution domain, $((n+1) \% n_x \neq 0)$, we set $\Delta_x = 1$.

Similarly, if $\hat{y}(0) = 0$, and the physical cell is not a lower boundary cell ($n \geq n_x$), we set $\Delta_y = -1$. And if $\hat{y}(0) = 1$, and the physical cell is not an upper boundary, ($n < n_x * (n_y - 1)$), then we set $\Delta_y = 1$.

Now the previous cell number, n_p will be given by

$$n_p = n + \Delta_y * n_x + \Delta_x \quad (\text{C.30})$$

For this case we will have that $\hat{x}(0) = 0$, $\hat{y}(0) = 0.061$, $n = 11$. So we find that $\Delta_y = 0$ and $\Delta_x = -1$. Consequently $n_p = 11 + 0 * 8 - 1 = 10$.

9. Inverse map of end point We are now back to step 3 above, and want to do an inverse map of the point P in the physical space in cell $n_p = 10$ corresponding to the starting point of the streamline in cell $n = 11$. It is possible to avoid the inverse mapping in this case. We will discuss this for two general quadrilaterals, $Q1$ and $Q2$. See Figure C.8. For

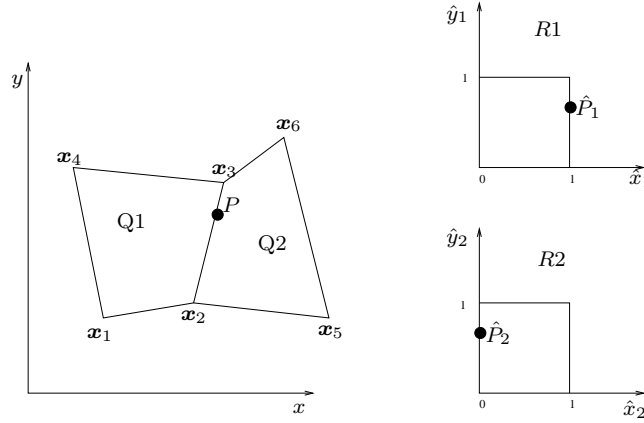


Figure C.8: Inverse map of point

the reference space $R1$ we have from (6.1), the bilinear transformation to $Q1$,

$$\begin{aligned} \mathbf{x}_{Q1}(\hat{\mathbf{x}}_{R1}) = & \mathbf{x}_1(1 - \hat{x}_1)(1 - \hat{y}_1) + \mathbf{x}_2\hat{x}_1(1 - \hat{y}_1) + \\ & \mathbf{x}_3\hat{x}_1\hat{y}_1 + \mathbf{x}_4(1 - \hat{x}_1)\hat{y}_1 \end{aligned} \quad (\text{C.31})$$

Similarly for $R2$ to $Q2$ we have

$$\begin{aligned} \mathbf{x}_{Q2}(\hat{\mathbf{x}}_{R2}) = & \mathbf{x}_2(1 - \hat{x}_2)(1 - \hat{y}_2) + \mathbf{x}_5\hat{x}_2(1 - \hat{y}_2) + \\ & \mathbf{x}_6\hat{x}_2\hat{y}_2 + \mathbf{x}_3(1 - \hat{x}_2)\hat{y}_2 \end{aligned} \quad (\text{C.32})$$

In the physical space, the edge of P can be given as,

$$\mathbf{r}(t) = (1 - t)\mathbf{x}_2 + t\mathbf{x}_3, \quad 0 \leq t \leq 1 \quad (\text{C.33})$$

Now consider the lines $\hat{x}_1 = 1$ and $\hat{x}_2 = 0$ in $R1$ and $R2$ respectively. Then by (C.31) and (C.32), we have

$$\mathbf{x}_{Q1}(1, \hat{y}_1) = (1 - \hat{y}_1)\mathbf{x}_2 + \hat{y}_1\mathbf{x}_3, \quad 0 \leq \hat{y}_1 \leq 1 \quad (\text{C.34})$$

$$\mathbf{x}_{Q2}(0, \hat{y}_2) = (1 - \hat{y}_2)\mathbf{x}_2 + \hat{y}_2\mathbf{x}_3, \quad 0 \leq \hat{y}_2 \leq 1 \quad (\text{C.35})$$

So both lines gets mapped to the edge of P . Set $\hat{P}_1 = (\hat{p}_{1x}, \hat{p}_{1y})$ and $\hat{P}_2 = (\hat{p}_{2x}, \hat{p}_{2y})$. If \hat{P}_1 and \hat{P}_2 are to be mapped to P then we must have that $\hat{p}_{1x} = 1$ and $\hat{p}_{2x} = 0$. Assuming $\hat{p}_{1y} \neq \hat{p}_{2y}$, we must also have that

$$\mathbf{x}_{Q_1}(1, \hat{p}_{1y}) \neq \mathbf{x}_{Q_2}(0, \hat{p}_{2y}) \quad (\text{C.36})$$

since by (C.31) and (C.32), $\mathbf{x}_{Q_1}(1, \hat{y})$ and $\mathbf{x}_{Q_2}(0, \hat{y})$ are one-to-one functions from R_1 and R_2 to the physical space such that $\mathbf{x}_{Q_1}(1, \hat{y}) \equiv \mathbf{x}_{Q_2}(0, \hat{y})$ for any value of \hat{y} . But since both \hat{P}_1 and \hat{P}_2 are to be mapped to P we have contradiction. Consequently $\hat{p}_{1y} = \hat{p}_{2y}$.

Generally, if we have two neighbouring quadrilaterals Q_1 and Q_2 and a point P lying on a common edge of Q_1 and Q_2 , the inverse bilinear transformation, \hat{P}_1 , of P corresponding to Q_1 does not have to be explicitly calculated when we already know the inverse transform, \hat{P}_2 , of P relative to Q_2 .

- 10. Backward tracing in cell 10** Using the argument of step 9, we must have that $\hat{P}_{10} = (1, 0.061)$. We can now return to step 5, and trace backward in cell 10. The only difference from cell 11 is that the known point \hat{P} now lies on the the boundary of the cell instead of inside the cell. So we don't have to apply forward tracing afterwards to get a complete streamline in cell 10. See Figure C.9. We find that

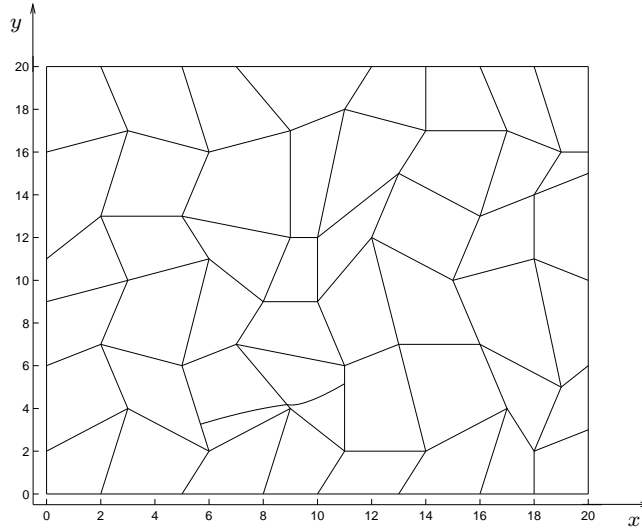


Figure C.9: Backward tracing in cell 10

$\hat{P}_0 = (0, 0.3158)$ for cell 10, and save the streamline in cell 10 in a *CellData* object that we adds to the *SLine* object.

11. Trace backward in cell 9 We return to step 8 to obtain the previous cell, and find that $n_p = 9$. We do the inverse map of step 9, for this cell, to discover that $\hat{P}_9 = (1, 0.3158)$. We then return to step 5 for backward tracing in cell 9. We will discuss this case in detail since cell 9 is a source cell. For this cell we have (see Figure C.10),

$$\hat{P} = (\hat{x}(T), \hat{y}(T)) = (1, 0.3158) \quad (\text{C.37})$$

$$u_A = -0.0127 \quad (\text{C.38})$$

$$u_B = 0.0490 \quad (\text{C.39})$$

$$v_C = -0.0123 \quad (\text{C.40})$$

$$v_D = 0.0371 \quad (\text{C.41})$$

$$G_x = u_B - u_A = 0.0617 \quad (\text{C.42})$$

$$G_y = v_D - v_C = 0.0494 \quad (\text{C.43})$$

$$u(T) = u_B = 0.0490 \quad (\text{C.44})$$

$$v(T) = G_y \hat{y}(T) + v_C = 0.0033 \quad (\text{C.45})$$

Now from (B.20) and CxA2 we have that $T_A = \infty$. From (B.21) and

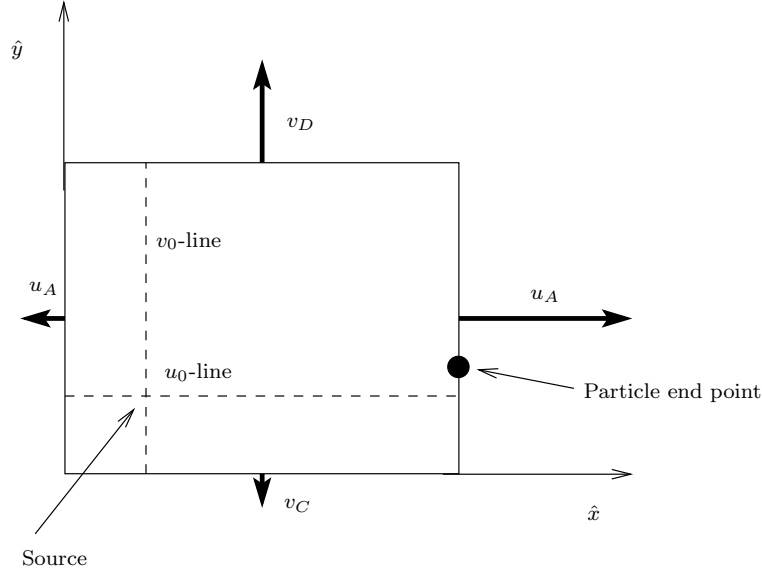


Figure C.10: Backward tracing in cell 9

CxB6 we have that $T_B = \infty$. From (B.22) and CyC2 we have that $T_C = \infty$, and from (B.23) and CyD3 we have that $T_D = \infty$.

Then from (B.39) and (B.40) we have that $\hat{P}_0 = (-\frac{u_A}{G_x}, -\frac{v_C}{G_y}) = (0.2055, 0.2497)$. Then using (B.44) and (B.45) to plot the streamline would only give the point P_0 , so we use (B.51) and (B.52), with $\hat{P}_0 = \hat{P}$. In Figure C.11 we have plotted the trajectory for $0 \leq t \leq 40$.

For times greater than $t = 40$ the points on the trajectory will lie so

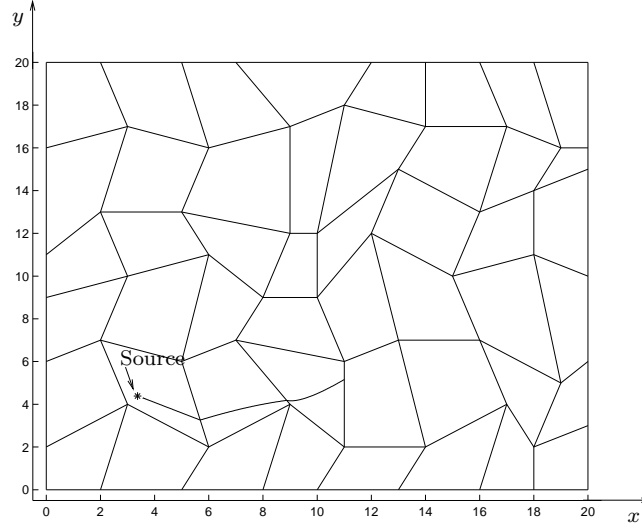


Figure C.11: Backward tracing in cell 9

close to the source that they can't be separated from each other.

12. Find the previous cell.

We now return to step 8 to find the previous streamline, but as $T = \infty$ there will be no previous cell for this streamline.

13. Forward tracing

We then return to the original point in cell 11, and start forward tracing from that point. But in step 6 we have already traced the streamline forward in cell 11. So we just have to find the next cell on the streamline. This is done in similar fashion to step 8, and we find the next cell to be number 12. The procedure should by now be clear. We continue tracing forward cell by cell until we reach the boundary of the domain or a sink. The whole streamline is plotted in figure C.12.

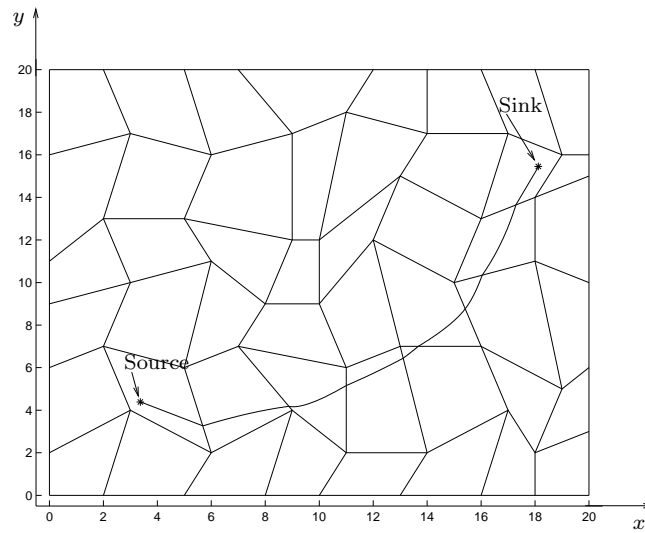


Figure C.12: The whole streamline

Appendix D

Determining if a point is inside a quadrilateral

When given a point inside the solution domain of a flow problem one often has to find the gridcell the point belongs to. In this thesis the point will be a starting point for a streamline and it will be necessary for the tracing algorithm to find the gridcell of the point.

Thus we will consider the problem of determining if a point, P , lies inside or outside an irregular quadrilateral, Q , in 2D. We will assume that Q is given as described in Definition 5.1, see Figure D.1. Next we define the term

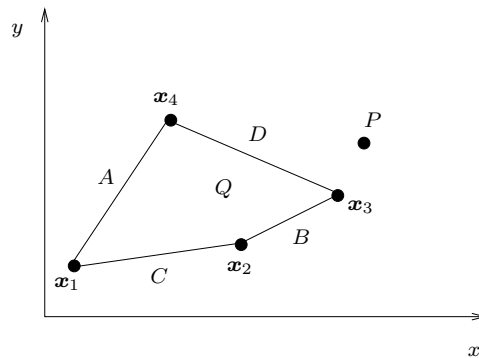


Figure D.1: Point inside quadrilateral?

inside more precisely,

Definition D.1 (Inside) *Given a point P and a quadrilateral Q , we say that P is inside Q if P lies on the boundary of Q or in the region enclosed by Q .*

We also need the following Definition

Definition D.2 (Interior of an edge) *The interior of an edge E with endpoints x_a and x_b are the points of E except x_a and x_b .*

We will need to consider the location of the corners of Q relative to each other.

Definition D.3 (Classifying corners) Consider a given corner $\mathbf{x}_i = (x_i, y_i)$ of Q , and its two neighbouring corners \mathbf{x}_{i+1} and \mathbf{x}_{i-1} , $i = 1, 2, 3, 4$. (We have here used $\mathbf{x}_0 \equiv \mathbf{x}_4$ and $\mathbf{x}_5 \equiv \mathbf{x}_1$.) We then say that \mathbf{x}_i is a corner of

Type-A: if $\{y_{i-1} < y_i < y_{i+1}\} \vee \{y_{i+1} < y_i < y_{i-1}\}$

Type-B: if $\{\{y_i > y_{i-1}\} \wedge \{y_i > y_{i+1}\}\} \vee \{\{y_i < y_{i-1}\} \wedge \{y_i < y_{i+1}\}\}$

Type-C: if $\{y_i = y_{i-1}\} \vee \{y_i = y_{i+1}\}$

See Figure D.2. Next, draw a horizontal ray, r , from P extending leftwards

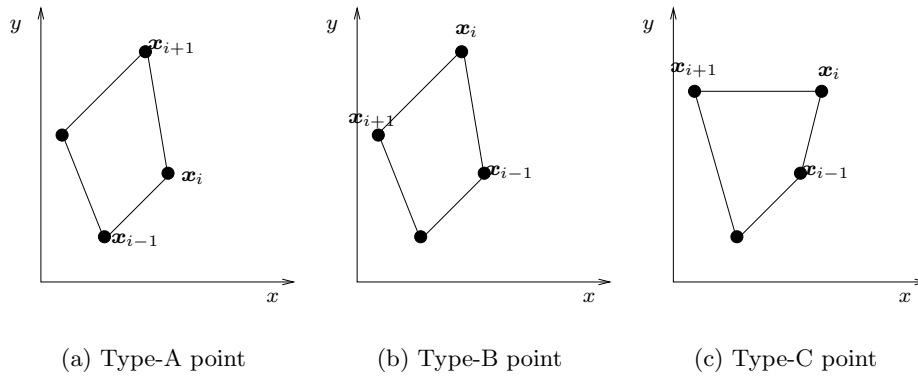


Figure D.2: Classifying points

to $x = -\infty$. See Figure D.3.

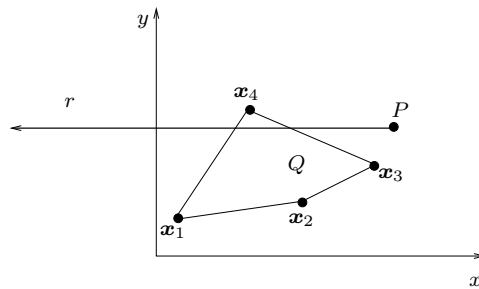


Figure D.3: Point inside quadrilateral?

Definition D.4 (Intersections) We say that the ray r has an intersection with the boundary of Q if it crosses the interior of an edge of Q or passes through a type-A corner of Q .

Now we can find out if P lies inside Q . If P is not on the boundary of Q then we count the number of intersections, i , that r makes with the boundary of Q . We then have the following result

Theorem D.1 *If P is not on the boundary of a convex quadrilateral, Q , and if $i = 1$, then P is inside Q , otherwise P is outside Q .*

Proof If $i = 0$ the point cannot be inside. Assume that $i = 1$ and assume that the intersection with the boundary of Q , I , is not a type-A point. Then the ray r will intersect the interior of an edge E of Q . We shall show that P is inside Q .

Let l_{ij} be the line containing \mathbf{x}_i and \mathbf{x}_j , and let α_i be the internal angle of Q corresponding to the corner \mathbf{x}_i . Define also l_{iP} as the line containing \mathbf{x}_i and P . Given a point $P = (\tilde{x}, \tilde{y})$, and a straight line, l , not horizontal or vertical, $y = ax + b$. We say that P lies *above* l if $\tilde{y} > a\tilde{x} + b$, and P lies *to the left* of l if $\tilde{x} < (\tilde{y} - b)/a$. We use similar definitions for *below* and *to the right*.

Without loss of generality, assume that $E = E_{12}$ (see Section 5.1 for terminology). We know that E_{12} is not horizontal, thus assume, still without loss of generality, that \mathbf{x}_1 lies above r and \mathbf{x}_2 lies below r . See Figure D.4. We will now consider possible locations for the point \mathbf{x}_3 .

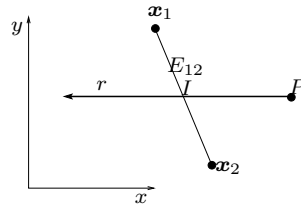


Figure D.4: The case of $i = 1$.

First, \mathbf{x}_3 cannot lie to the left of l_{12} . Because then either E_{23} crosses r , and then $i > 1$ or else \mathbf{x}_4 has to be to the right of l_{1P} to avoid that E_{41} intersects r . See Figure D.5. But then $\alpha_2 > 180^\circ$ and Q is not convex.

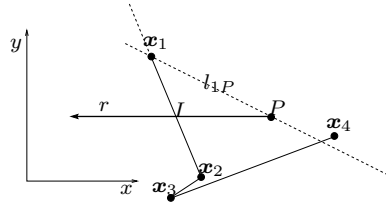


Figure D.5: The case of $i = 1$.

Thus we can assume that \mathbf{x}_3 lies to the right of l_{12} . But then \mathbf{x}_3 has to lie to the right of l_{2P} to avoid intersection with r , and then it is not possible

to choose \mathbf{x}_4 such that E_{41} does not cross r and P is outside Q at the same time. Thus P must be inside Q .

Now assume $i = 1$ and that I is of type-A. See Figure D.6. We can

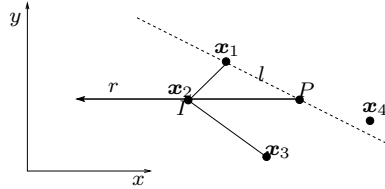
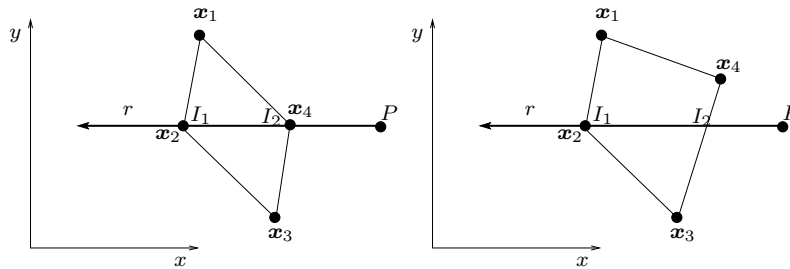


Figure D.6: The case of $i = 1$ and \mathbf{x}_2 is of type-A

assume without loss of generality that $\mathbf{x}_1, \mathbf{x}_2$ and \mathbf{x}_3 is located as in Figure D.6. Then \mathbf{x}_4 has to be to the right of l_{1P} to avoid that E_{41} intersects r . But then E_{41} will enclose P inside Q .

Next assume $i = 2$. And denote the intersection points by I_1 and I_2 . If at least one of I_1 and I_2 are of type-A. Then we have the situation in Figure D.7(a) or D.7(b). Then certainly P does not lie inside Q . So assume that



(a) The case of $i = 2$ and both I_1 and I_2 are type-A

(b) The case of $i = 2$ and I_1 is type-A

Figure D.7: The case $i = 2$.

$i = 2$ and that neither I_1 nor I_2 is of type-A, and that I_1 is contained in E_{12} then if the edge containing I_2 is not connected to E_{12} it must be the edge E_{34} , and we have the situation on Figure D.8. In this case P will not lie inside Q . If on the other hand the edge of I_2 is joined in \mathbf{x}_2 to E_{12} we have the situation in Figure D.9. Then \mathbf{x}_4 has to lie in the region, R , (see Figure D.9) enclosed by l_{12}, l_{23} and l_{13} . Otherwise Q will not be convex. Thus P is not inside Q . This means that for a convex Q we can have at most $i = 2$. ■

We will now use the theorem to solve the problem. At first, we should check if $P = (X, Y)$ lies on the boundary of Q . We start by checking if P lies on one of the cornerpoints of Q . Then we check if it is contained in the

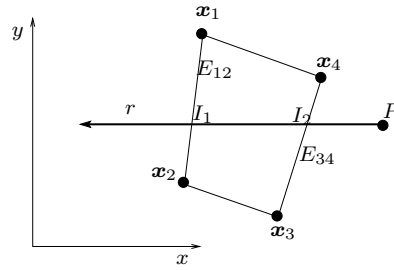


Figure D.8: The case of $i = 2$ and crossing edges not connected.

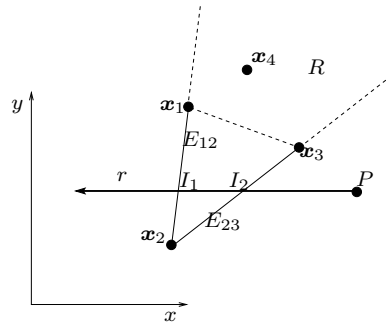


Figure D.9: The case of $i = 2$ and crossing edges connected.

interior of an edge E , not horizontal, of Q . We can then at the same time start checking for interior intersections. See Figure D.10. The slope of the

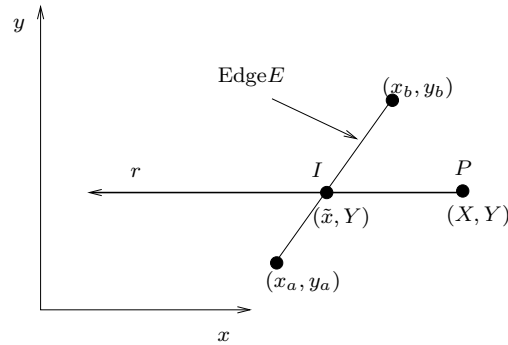


Figure D.10: We want to find out if the ray r intersect the edge E .

edge is given by

$$m = \frac{y_b - y_a}{x_b - x_a} \tag{D.1}$$

and the equation for the edge is

$$y = m(x - x_a) + y_b \tag{D.2}$$

Let $P = (X, Y)$. Then the intersection point is $I = (\tilde{x}, Y)$ where \tilde{x} is given by (D.2) by inserting $y = Y$,

$$\tilde{x} = \frac{Y - y_a}{m} + x_a \quad (\text{D.3})$$

Then if $X = \tilde{x}$ P lies on the boundary of Q otherwise if $X > \tilde{x}$ the ray will intersect E provided $y_a < Y < y_b$ or $y_b < Y < y_a$.

If $E = E_{ij}$ is horizontal then P is contained in E if $x_i < X < x_j$ or $x_j < X < x_i$. Now assume P is not on the boundary, it remains to check for type-A intersections. If a cornerpoint $\mathbf{x}_i = (x_i, y_i)$ of Q is type-A, then we will have type-A intersection if $Y = y_i$ and $X > x_i$.

A complete C++ algorithm is given in Appendix N.1.1.

Appendix E

The inverse bilinear transformation

In this section we will consider the inverse transform, of (6.1), $\hat{\mathbf{x}}(\mathbf{x})$. That is, we want to find the point in the reference space, \mathcal{R} , coresponding to a given point in the physical space, \mathcal{P} . We denote the quadrilateral in \mathcal{P} corresponding to \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 and \mathbf{x}_4 by $Q(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$, and let \hat{Q} denote the corresponding unit square in \mathcal{R} .

In \mathcal{P} we shall only consider points within Q , whereas in \mathcal{R} we shall only consider points within \hat{Q} . We now rewrite (6.1) as

$$a_x \hat{x} \hat{y} = b_x \hat{x} + c_x \hat{y} + d_x \quad (\text{E.1a})$$

$$a_y \hat{x} \hat{y} = b_y \hat{x} + c_y \hat{y} + d_y \quad (\text{E.1b})$$

where

$$a_x = x_1 - x_2 + x_3 - x_4 \quad (\text{E.2a})$$

$$b_x = x_1 - x_2 \quad (\text{E.2b})$$

$$c_x = x_1 - x_4 \quad (\text{E.2c})$$

$$d_x = x - x_1 \quad (\text{E.2d})$$

$$a_y = y_1 - y_2 + y_3 - y_4 \quad (\text{E.2e})$$

$$b_y = y_1 - y_2 \quad (\text{E.2f})$$

$$c_y = y_1 - y_4 \quad (\text{E.2g})$$

$$d_y = y - y_1 \quad (\text{E.2h})$$

We also set

$$e_x = x_3 - x_4 \quad (\text{E.3a})$$

$$e_y = y_3 - y_4 \quad (\text{E.3b})$$

$$r = c_x a_y - c_y a_x \quad (\text{E.3c})$$

$$s = b_x a_y - b_y a_x = b_x e_y - b_y e_x \quad (\text{E.3d})$$

$$t = d_x a_y - d_y a_x \quad (\text{E.3e})$$

E.1 Case 1

Assuming

$$a_x \neq 0 \quad (\text{E.4})$$

we can solve for equation (E.1a) for $\hat{x}\hat{y}$,

$$\hat{x}\hat{y} = \frac{d_x + b_x \hat{x} + c_x \hat{y}}{a_x} \quad (\text{E.5})$$

Inserting this in equation (E.1b) and collecting \hat{x} terms, we get

$$s\hat{x} = -r\hat{y} - t \quad (\text{E.6})$$

Assume

$$s \neq 0 \quad (\text{E.7})$$

then we can solve equation (E.6) for \hat{x} ,

$$\hat{x} = -\frac{r\hat{y}}{s} - \frac{t}{s} \quad (\text{E.8})$$

Inserting this back in equation (E.1a) and collecting \hat{y} -terms, we get

$$-ra_x \hat{y}^2 + (rb_x - ta_x - c_x s)\hat{y} + tb_x - d_x s = 0 \quad (\text{E.9})$$

If $r \neq 0$, equation (E.9) is on the form

$$A\hat{y}^2 + B\hat{y} + C = 0 \quad (\text{E.10})$$

where $A \neq 0$. The solutions are

$$\hat{y}_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (\text{E.11})$$

$$\hat{y}_2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad (\text{E.12})$$

(We will not consider the case $B^2 - 4AC < 0$. For convex Q this should not happen.) Then we choose the solution of equation (E.10) which satisfies

$$0 \leq \hat{y} \leq 1 \quad (\text{E.13})$$

Now \hat{x} is found from equation (E.8).

E.2 Case 2

Still using assumptions (E.4) and (E.7) from Case 1, and furthermore assuming $r = 0$ in (E.8) then

$$\hat{x} = -\frac{t}{s} \quad (\text{E.14})$$

and from (E.9) we have

$$\hat{y} = -\frac{tb_x - d_x s}{ta_x + c_x s} \quad (\text{E.15})$$

If $ta_x + c_x s = 0$ then it follows from equation (E.9) that $tb_x - d_x s = 0$, and then equation (E.15) gives that \hat{y} is undefined. This should not happen when $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and \mathbf{x}_4 represents a quadrilateral and \mathbf{x} is a point inside the quadrilateral.

E.3 Case 3

Starting from equation (E.6) in Case 1 and assuming $s = 0$, we note from the same equation that r should not be zero in this case, unless $t = 0$. Assuming $r \neq 0$, we get

$$\hat{y} = -\frac{t}{r} \quad (\text{E.16})$$

and then \hat{x} can be found from (E.5) as

$$\hat{x} = \frac{c_x t - r d_x}{b_x r + a_x t} \quad (\text{E.17})$$

If $b_x r + a_x t = 0$ then also $c_x t - r d_x = 0$, and then equation (E.17) gives that \hat{x} is undefined. This should not happen as noted in Case 2.

E.4 Case 4

If $a_x = 0$ in equation (E.1a) we can write that equation as

$$c_x \hat{y} = -b_x \hat{x} - d_x \quad (\text{E.18})$$

and if $c_x \neq 0$ we can solve for \hat{y} ,

$$\hat{y} = \frac{-b_x \hat{x} - d_x}{c_x} \quad (\text{E.19})$$

So substituting equation (E.19) in (E.1b) gives

$$-a_y b_x \hat{x}^2 + (c_y b_x - b_y c_x - a_y d_x) \hat{x} + c_y d_x - d_y c_x = 0 \quad (\text{E.20})$$

Now, if either $a_y = 0$ or $b_x = 0$, we get

$$\hat{x} = \frac{d_y c_x - c_y d_x}{c_y b_x - b_y c_x - a_y d_x} \quad (\text{E.21})$$

Note that $c_y b_x - b_y c_x - a_y d_x$ should not be zero, because then \hat{x} is undefined (see case 2). When \hat{x} is given by equation (E.21), \hat{y} is given by equation (E.19).

Assuming $a_y \neq 0$ and $b_x \neq 0$ equation (E.20) is on the form

$$A\hat{x}^2 + B\hat{x} + C = 0 \quad (\text{E.22})$$

where $A \neq 0$. The solutions are

$$\hat{x}_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (\text{E.23})$$

$$\hat{x}_2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad (\text{E.24})$$

We will choose the solution of equation (E.22) which satisfies

$$0 \leq \hat{x} \leq 1 \quad (\text{E.25})$$

Now \hat{y} is found from equation (E.19).

E.5 Case 5

If $c_x = 0$ in equation (E.18) we get from that equation,

$$b_x \hat{x} = -d_x \quad (\text{E.26})$$

Note that b_x cannot be zero at same time as c_x is zero because then $x_1 = x_2 = x_4$ and we have a triangle and not a quadrilateral.

Consequently,

$$\hat{x} = -\frac{d_x}{b_x} \quad (\text{E.27})$$

Substituting this in equation (E.1b) we get

$$(a_y d_x + c_y b_x) \hat{y} = b_y d_x - d_y b_x \quad (\text{E.28})$$

If $a_y d_x + c_y b_x = 0$ then \hat{y} is undefined (see Case 2), otherwise

$$\hat{y} = \frac{b_y d_x - d_y b_x}{a_y d_x + c_y b_x} \quad (\text{E.29})$$

Appendix F

The flux over an edge of a quadrilateral

In this appendix we will derive an expression for the flux for a velocity field, \mathbf{q} , over an edge of a quadrilateral. We will assume that the endpoints of the edge is $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$. The flux over the edge, E , is given by

$$f_E = \int_E \mathbf{q} \cdot \mathbf{n}_E ds \quad (\text{F.1})$$

see Figure F.

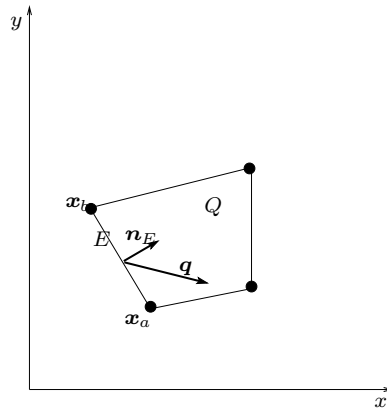


Figure F.1: Calculation of flux across the edge E .

A vector parallel to E is given by

$$\mathbf{t}_E = \mathbf{b} - \mathbf{a} = (b_1 - a_1, b_2 - a_2) \quad (\text{F.2})$$

A unit normal vector to the edge E is therefore given by

$$\mathbf{n}_E \cdot \mathbf{t}_E = 0 \quad \wedge \quad \|\mathbf{n}_E\| = 1 \quad (\text{F.3})$$

This gives that

$$\mathbf{n}_E = \pm \frac{1}{\|\mathbf{t}_E\|} (a_2 - b_2, b_1 - a_1) \quad (\text{F.4})$$

The edge E is given by

$$\mathbf{r}(t) = (1 - t)\mathbf{a} + t\mathbf{b}, \quad 0 \leq t \leq 1 \quad (\text{F.5})$$

From (F.1) we now have

$$f_E = \int_E \mathbf{q} \cdot \mathbf{n}_E ds = \pm \int_0^1 \left(\mathbf{q}(\mathbf{r}(t)) \cdot \frac{1}{\|\mathbf{t}_E\|} (a_2 - b_2, b_1 - a_1) \right) \left\| \frac{d\mathbf{r}}{dt} \right\| dt \quad (\text{F.6})$$

Since $\left\| \frac{d\mathbf{r}}{dt} \right\| = \|\mathbf{t}_E\|$ this reduces to

$$f_E = \pm \int_0^1 (\mathbf{q}(\mathbf{r}(t))) \cdot (a_2 - b_2, b_1 - a_1) dt \quad (\text{F.7})$$

Appendix G

The uniform velocity field, additional computations

G.1 Flux computations

In this Appendix, we will calculate edge fluxes for a given quadrilateral, Q , when we are given a uniform velocity field $\mathbf{q} = (q_x, q_y)$, where q_x and q_y are constants.

From Appendix F we have that the flux of \mathbf{q} over an edge E with endpoints $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$, of Q is given by

$$f_E = \pm \int_0^1 (\mathbf{q} \cdot (a_2 - b_2, b_1 - a_1)) dt = \pm \mathbf{q} \cdot (a_2 - b_2, b_1 - a_1) = \mathbf{q} \cdot \mathbf{n}_E \quad (\text{G.1})$$

since \mathbf{q} is constant.

Remember from Section 8.4, that in order to calculate the correct flux for a given edge a certain direction of the normal had to be chosen depending on the edge. For an A -edge the normal should be directed inwards relative to Q . See Figure G.1. So how do we choose the correct sign of \mathbf{n}_E ? We will not answer this question for a general grid, but assume that

$$\mathbf{n}_A = (y_4 - y_1, x_1 - x_4) \quad (\text{G.2a})$$

$$\mathbf{n}_B = (y_3 - y_2, x_2 - x_3) \quad (\text{G.2b})$$

$$\mathbf{n}_C = (y_1 - y_2, x_2 - x_1) \quad (\text{G.2c})$$

$$\mathbf{n}_D = (y_4 - y_3, x_3 - x_4) \quad (\text{G.2d})$$

where $\mathbf{x}_1 = (x_1, y_1)$, $\mathbf{x}_2 = (x_2, y_2)$, $\mathbf{x}_3 = (x_3, y_3)$, and $\mathbf{x}_4 = (x_4, y_4)$. This assumption holds on a “normal” grid, e.g. Figure G.2.

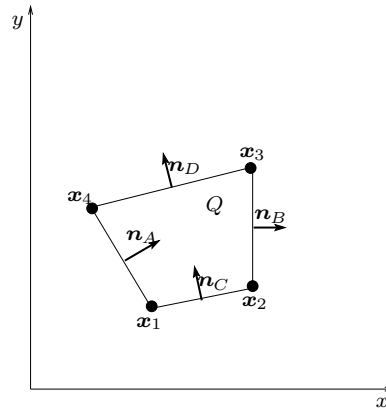


Figure G.1: Gridcell

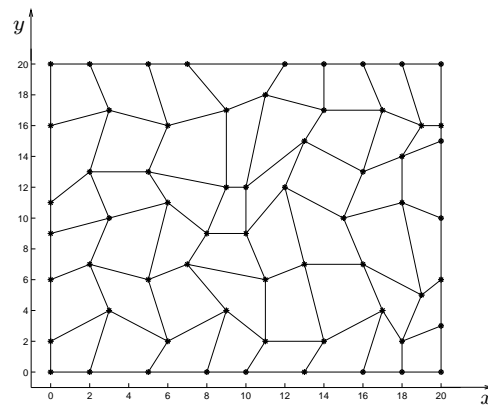


Figure G.2: Sample grid

Using (G.2) in (G.1), we now have

$$f_A = \mathbf{q} \cdot (y_4 - y_1, x_1 - x_4) = q_x(y_4 - y_1) + q_y(x_1 - x_4) \quad (\text{G.3a})$$

$$f_B = \mathbf{q} \cdot (y_3 - y_2, x_2 - x_3) = q_x(y_3 - y_2) + q_y(x_2 - x_3) \quad (\text{G.3b})$$

$$f_C = \mathbf{q} \cdot (y_1 - y_2, x_2 - x_1) = q_x(y_1 - y_2) + q_y(x_2 - x_1) \quad (\text{G.3c})$$

$$f_D = \mathbf{q} \cdot (y_4 - y_3, x_3 - x_4) = q_x(y_4 - y_3) + q_y(x_3 - x_4) \quad (\text{G.3d})$$

G.2 Analytical tracing

In this section we will use the analytically computed fluxes in the previous section to trace streamlines in \mathcal{R} and then map these streamlines back to \mathcal{P} to see if they correspond to the known solution, i.e. straight lines. See Figure G.3 for notation. Let the starting point be $\hat{\mathbf{x}}_0 = (\hat{x}_0, \hat{y}_0)$, and assume

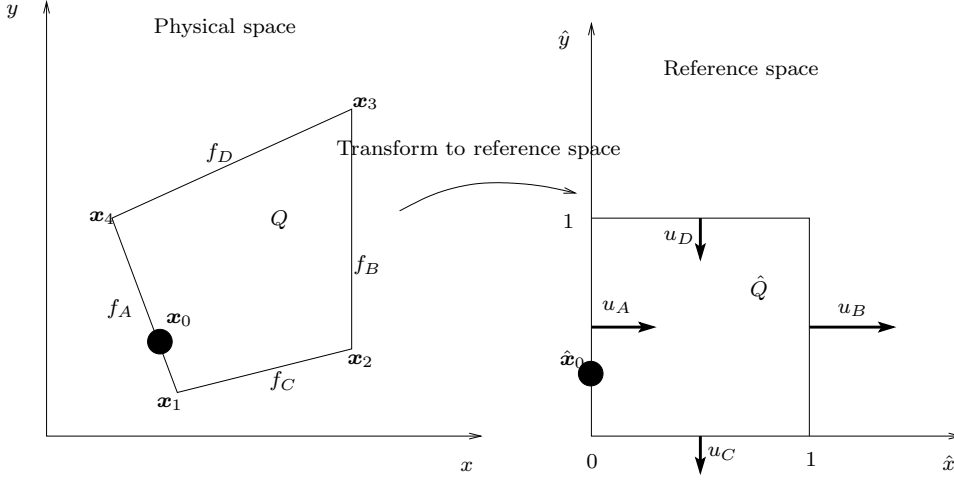


Figure G.3: Map of fluxes to the reference space.

the travel time T has been found, then from section B.2, we find that the trajectory is given by

$$\hat{x}(t) = \begin{cases} \frac{1}{G_x} ((G_x \hat{x}_0 + u_A)e^{G_x t} - u_A), & G_x \neq 0, & 0 \leq t \leq T \\ u_A t + \hat{x}_0, & G_x = 0, & 0 \leq t \leq T \end{cases} \quad (\text{G.4})$$

$$\hat{y}(t) = \begin{cases} \frac{1}{G_y} ((G_y \hat{y}_0 + v_C)e^{G_y t} - v_C), & G_y \neq 0, & 0 \leq t \leq T \\ v_C t + \hat{y}_0, & G_y = 0, & 0 \leq t \leq T \end{cases} \quad (\text{G.5})$$

Due to the uniform velocity field there are no sources or sinks within the cell. Mass conservation then implies

$$f_A - f_B + f_C - f_D = 0 \quad (\text{G.6})$$

Now the cell face velocities u_A, u_B, v_C, v_D in reference space is given by

$$u_A = f_A/J, \quad u_B = f_B/J, \quad v_C = f_C/J, \quad v_D = f_D/J \quad (\text{G.7})$$

where J is the area of the physical cell. From (G.6) we have

$$f_B/J - f_A/J = f_C/J - f_D/J \quad (\text{G.8})$$

$$\Rightarrow u_B - u_A = -(u_D - u_C) \quad (\text{G.9})$$

$$\Rightarrow G_x = -G_y \quad (\text{G.10})$$

Assume $G_x \neq 0$ and solve Equation (G.4) for t :

$$t = \frac{1}{G_x} \ln \left(\frac{G_x \hat{x} + u_A}{G_x \hat{x}_0 + u_A} \right) \quad (\text{G.11})$$

Assume that $G_y \neq 0$ and substitute (G.11) in (G.5),

$$\hat{y}(\hat{x}) = \frac{1}{G_y} \left\{ e^{\left[\frac{G_y}{G_x} \ln \left(\frac{G_x \hat{x} + u_A}{G_x \hat{x}_0 + u_A} \right) \right]} (G_y \hat{y}_0 + v_C) - v_C \right\} \quad (\text{G.12})$$

$$\text{(by (G.10))} = \frac{1}{-G_x} \left\{ e^{\left[-\ln \left(\frac{G_x \hat{x} + u_A}{G_x \hat{x}_0 + u_A} \right) \right]} (-G_x \hat{y}_0 + v_C) - v_C \right\} \quad (\text{G.13})$$

$$= \frac{G_x \hat{x}_0 \hat{y}_0 - \hat{x}_0 v_C + \hat{y}_0 u_A + \hat{x} v_C}{\hat{x} G_x + u_A} \quad (\text{G.14})$$

We now use the relations in (G.3) to get

$$u_A = \frac{f_A}{J} = \frac{1}{J} (q_x(y_4 - y_1) + q_y(x_1 - x_4)) \quad (\text{G.15})$$

$$u_B = \frac{f_B}{J} = \frac{1}{J} (q_x(y_3 - y_2) + q_y(x_2 - x_3)) \quad (\text{G.16})$$

$$v_C = \frac{f_C}{J} = \frac{1}{J} (q_x(y_1 - y_2) + q_y(x_2 - x_1)) \quad (\text{G.17})$$

$$G_x = \frac{1}{J} (q_x(y_1 + y_3 - y_2 - y_4) + q_y(x_2 + x_4 - x_1 - x_3)) \quad (\text{G.18})$$

We would now like to insert these relations in (G.14). But it turns out the the computations become quite involved, so it was given up. Let's instead assume that $q_x = 1$ and $q_y = 0$. Then insert (G.15), (G.16), (G.17) and (G.18) in (G.14) and simplify the result to get

$$\hat{y}(\hat{x}) = \frac{\hat{x}_0 \hat{y}_0 (y_1 + y_3 - y_2 - y_4) + \hat{x}_0 (y_2 - y_1) + \hat{y}_0 (y_4 - y_1) + \hat{x} (y_1 - y_2)}{\hat{x} (y_1 + y_3 - y_2 - y_4) + y_4 - y_1} \quad (\text{G.19})$$

Now we want to transform this curve in reference space back to physical space using the transform (6.1). It turns out that we need only y -component of the transformation

$$y(\hat{x}, \hat{y}) = y_1(1 - \hat{x})(1 - \hat{y}) + y_2 \hat{x}(1 - \hat{y}) + y_3 \hat{x} \hat{y} + y_4(1 - \hat{x}) \hat{y} \quad (\text{G.20})$$

Substituting (G.19) in (G.20) we get after simplifying

$$y(x) = \hat{x}_0 \hat{y}_0 (y_1 + y_3 - y_2 - y_4) + \hat{x}_0 (y_2 - y_1) + \hat{y}_0 (y_4 - y_1) + y_1 \quad (\text{G.21})$$

so $y(x)$ is constant, i.e: a horizontal line, and streamlines should be straight horizontal lines.

G.3 Analytical TOF

For a straight streamline of length s and a constant velocity field, $\mathbf{q} = (1, \tan(\alpha))$, the TOF, τ , is given by

$$\tau = \frac{s}{|\mathbf{q}|} = \frac{s}{\sqrt{1 + \tan^2(\alpha)}} \quad (\text{G.22})$$

see Figure G.4.

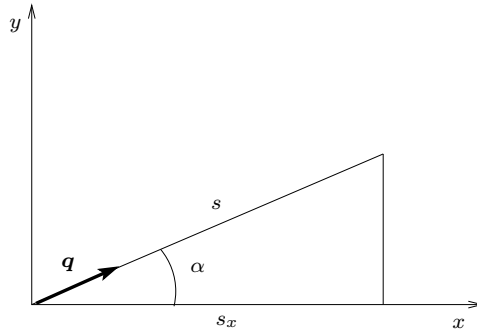


Figure G.4: Calculation of TOF for the streamline s .

For $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ and \mathbf{q}_4 (see Section 10.1) corresponding to $\alpha = 10^\circ, \alpha = 20^\circ, \alpha = 30^\circ$ and $\alpha = 40^\circ$ respectively, we know that $s_x = 19$, see Figure G.4. Thus $s = s_x / \cos(\alpha)$, and inserting this in (G.22), we find that $\tau_i = 19$, $i = 1, 2, 3, 4$.

For $\mathbf{q}_5, \mathbf{q}_6, \mathbf{q}_7$ and \mathbf{q}_8 corresponding to $\alpha = 50^\circ, \alpha = 60^\circ, \alpha = 70^\circ$ and $\alpha = 80^\circ$ respectively, we find that $s = 19 / \sin(\alpha)$, and substituting this in (G.22), we find that $\tau_5 = 15.9429, \tau_6 = 10.9697, \tau_7 = 6.9154, \tau_8 = 3.3502$.

Appendix H

The circular velocity field, additional computations

H.1 Computation of fluxes

In this Appendix we will calculate the flux of the velocity \mathbf{q} given in (10.1), over an edge E of a quadrilateral Q in the simulation grid. See Figure H.1. The flux over the edge E is by (H.1)

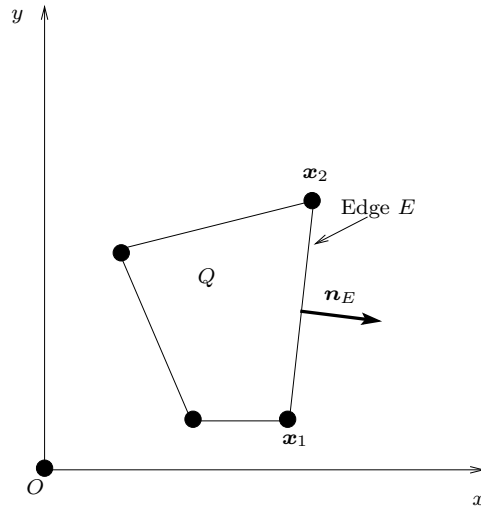


Figure H.1: Calculating the flux over the edge E .

$$f_E = \pm \int_0^1 (\mathbf{q}(\mathbf{r}(t))) \cdot (a_2 - b_2, b_1 - a_1) dt = \int_0^1 (\mathbf{q}(\mathbf{r}(t))) \cdot \mathbf{n}_E \quad (\text{H.1})$$

where the endpoints of the edge E is $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$. E can be represented as

$$\mathbf{r}(t) = (1 - t)\mathbf{a} + t\mathbf{b} = [(1 - t)a_1 + tb_1, (1 - t)a_2 + tb_2], \quad 0 \leq t \leq 1 \quad (\text{H.2})$$

We will not discuss how to find the correct normal, \mathbf{n}_E . The the problem has been discussed in other sections.

Substituting (10.1) for $\mathbf{q}(x, y)$, (I.3) can be written as

$$f_E = \int_0^1 \frac{n_1(y_0 - (1 - t)y_1 - ty_2) + n_2((1 - t)x_1 + tx_2 - x_0)}{\sqrt{((1 - t)x_1 + tx_2 - x_0)^2 + ((1 - t)y_1 + ty_2 - y_0)^2}} dt \quad (\text{H.3})$$

This integral is complicated to evaluate analytically, so we will use numerical integration. The integral can be written

$$\int_0^1 f(t) dt \quad (\text{H.4})$$

where $f(t)$ is given by

$$f(t) = \frac{n_1(y_0 - (1 - t)y_1 - ty_2) + n_2((1 - t)x_1 + tx_2 - x_0)}{\sqrt{((1 - t)x_1 + tx_2 - x_0)^2 + ((1 - t)y_1 + ty_2 - y_0)^2}} \quad (\text{H.5})$$

Observe that $f(t)$ is singular at the origin, O . We will assume that the origin is not placed on an edge of the grid or in the vicinity of an edge. Then $f(t)$ is well behaved, i.e., $|f''(t)| \leq K$ for some constant K “not too big”. In this case a simple midpoint rule can be used for numerical integration.

The method is based on a partition of $[0, 1]$ into n equal subintervals and then forming a Riemann sum of the areas of rectangles whose heights are taken at the midpoints of the n subintervals. See figure H.2. If $h = 1/n$, let

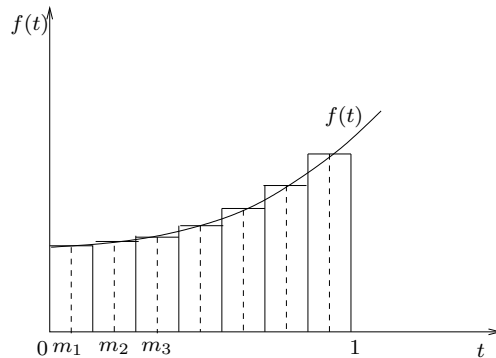


Figure H.2: The midpoint rule.

$m_j = (j - \frac{1}{2})h$ for $1 \leq j \leq n$. An approximation to (H.4) is then given by

$$M_n = h \sum_{j=1}^n f(m_j) \quad (\text{H.6})$$

An error estimate for this approximation is given by

$$\left| \int_0^1 f(t) dt - M_n \right| \leq \frac{K}{24n^2} \quad (\text{H.7})$$

where

$$K = \max_{0 \leq t \leq 1} (|f''(t)|) \quad (\text{H.8})$$

Appendix I

The y -dependent velocity field

In this Appendix we will calculate the flux of the velocity field $\mathbf{q} = (1, y)$ over an edge, E , of a quadrilateral, Q . From (H.1) the flux is given by

$$f_E = \pm \int_0^1 (\mathbf{q}(\mathbf{r}(t))) \cdot (a_2 - b_2, b_1 - a_1) dt = \int_0^1 (\mathbf{q}(\mathbf{r}(t))) \cdot \mathbf{n}_E dt \quad (\text{I.1})$$

where $\mathbf{n}_E = (n_1, n_2)$, $\mathbf{r}(t)$ is a parametric representation for the edge E , with endpoints $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$.

$\mathbf{r}(t)$ is given by

$$\mathbf{r}(t) = (1 - t)\mathbf{a} + t\mathbf{b} = ((1 - t)a_1 + tb_1, (1 - t)a_2 + tb_2) \quad (\text{I.2})$$

Then

$$f_E = \int_0^1 (1, (1 - t)a_2 + tb_2) \cdot (n_1, n_2) dt = \frac{n_2(a_2 + b_2)}{2} + n_1 \quad (\text{I.3})$$

Appendix J

Calculation of time-of-flight

J.1 The single cell

Consider the quadrilateral, Q , given in figure J.1, where $\mathbf{x}_1 = (x_1, y_1)$,

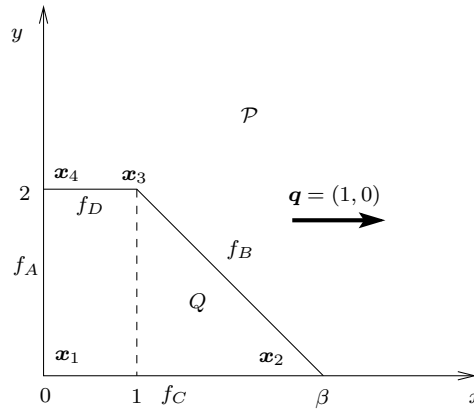


Figure J.1: The quadrilateral used in the test.

$\mathbf{x}_2 = (x_2, y_2)$, $\mathbf{x}_3 = (x_3, y_3)$, $\mathbf{x}_4 = (x_4, y_4)$.

According to (G.3) the fluxes are given by

$$f_A = y_4 - y_1 = 2 \quad (\text{J.1})$$

$$f_B = y_3 - y_2 = 2 \quad (\text{J.2})$$

$$f_C = y_1 - y_2 = 0 \quad (\text{J.3})$$

$$f_D = y_4 - y_3 = 0 \quad (\text{J.4})$$

Now tracing is done in \mathcal{R} . See Figure J.2, where according to (8.57), $u_A = u_B = 2/J_m$. The velocity in \mathcal{R} is given by

$$\frac{dx}{dt} = u_A \quad (\text{J.5})$$

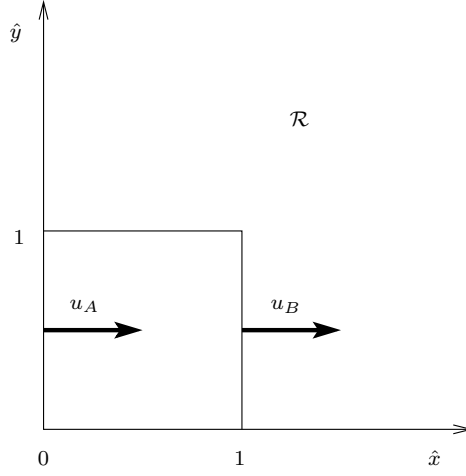


Figure J.2: Tracing in reference space.

since $u_A = u_B$. Then $t = x/u_A$ and the travel time, T , for a particle starting at $\hat{x} = 0$ is $T = 1/u_A = J_m/2$.

The bilinear transformation, (6.1), from \mathcal{R} to \mathcal{P} is for this case,

$$x = \beta\hat{x}(1 - \hat{y}) + \hat{x}\hat{y} \quad (\text{J.6})$$

$$y = 2\hat{x}\hat{y} + 2(1 - \hat{x})\hat{y} = 2\hat{y} \quad (\text{J.7})$$

So

$$D = \begin{bmatrix} (1 - \beta)\hat{y} + \beta & (1 - \beta)\hat{x} \\ 0 & 2 \end{bmatrix} \quad (\text{J.8})$$

and

$$J = 2\hat{y}(1 - \beta) + 2\beta \quad (\text{J.9})$$

J.2 The skew grid

We calculate the travel time for the particles p_1 , p_2 , p_3 and p_4 as defined in Section 10.5.2. First consider the first or bottom row of the grid. See Figure J.3. The trapezoids Q_i , $i = 1, \dots, 10$ all have two parallel sides of length d_1 and d_2 respectively. Consequently they all have the same area and thus by (6.11), the same midpoint Jacobi determinant, J_m . In this grid, $d_1 = 2$ and $d_2 = 1.95$ thus $J_m = (d_1 + d_2)/2 = 1.975$. A similar calculation as in the section above shows then that the travel time for any particle starting on the left boundary of Q_i will use $T = J_m$ seconds to exit the cell at right boundary.

Now particle p_1 has to pass all the cells Q_i to enter the line at $x = 20$, consequently it will use $T = T_1 = 10 * J_m = 19.75$ seconds to reach the line.

The particle p_2 will also pass all the ten cells, and in addition it will travel the distance d_3 in cell 11, given by $d_3 = 20 - 10d_2 = 0.5$. Consequently it

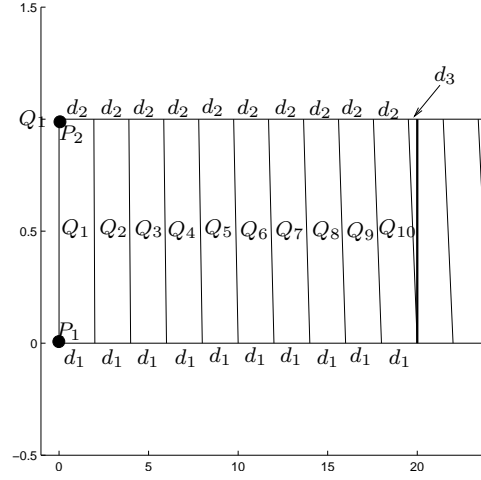


Figure J.3: The first ten quadrilaterals of the first row of the grid.

will travel a distance \hat{d}_3 in \mathcal{R} for cell 11, where $\hat{d}_3 = d_3/d_2 \approx 0.264$, and the time used in cell 11 is then $\Delta t_2 = \hat{d}_3 J_m \approx 0.5$. Then $T_2 = T_1 + \Delta t_2 = 20.25$.

For the top row of the grid, we have that $d_1 = 1.05$, $d_2 = 1$, $J_m = (d_1 + d_2)/2 = 1.025$. We see that particle p_4 has to pass twenty cells to reach l , so $T_4 = 20J_m = 20.5$. Particle p_3 passes 19 cells, and in addition a distance d_3 in the rightmost cell, given by $d_3 = 20 - 19d_2 = 0.05$. Thus $\hat{d}_3 = d_3/d_2 \approx 0.048$, and $\Delta t_3 = \hat{d}_3 J_m \approx 0.05$, and $T_3 = 19J_m + \Delta t_3 = 19.48$.

Now assume that trajectory of p_2 will lie in the cells in second row of the grid instead of the first row. Then $d_1 = 1.95$, $d_2 = 1.9$, $J_m = (d_1 + d_2)/2 = 1.925$. In this case p_2 will pass the first 10 cells in row two, and in addition a distance d_3 in 11th cell of row two, where $d_3 = 20 - 10d_1 = 0.5$. As above we find that $\hat{d}_3 = d_3/d_1 \approx 0.256$, and the time used in this cell is $\Delta t_2 \approx 0.0494$. Thus $\tilde{T}_2 = 10J_m + \Delta t_2 = 19.3$. Comparing this to T_2 above, we see that the time-of-flight is discontinuous along the trajectory of p_2 . This is also true for the other particles, since they all have trajectories on interfaces between cells of the grid with different midpoint Jacobi determinant.

Appendix K

Generation of a random convex grid.

In this Appendix we will describe how to generate a random convex grid. Such a grid can be used to test the streamline method.

We will generate the grid such that the boundary of the grid is rectangle. See Figure K.1. The grid will be generated as a random perturbation of a

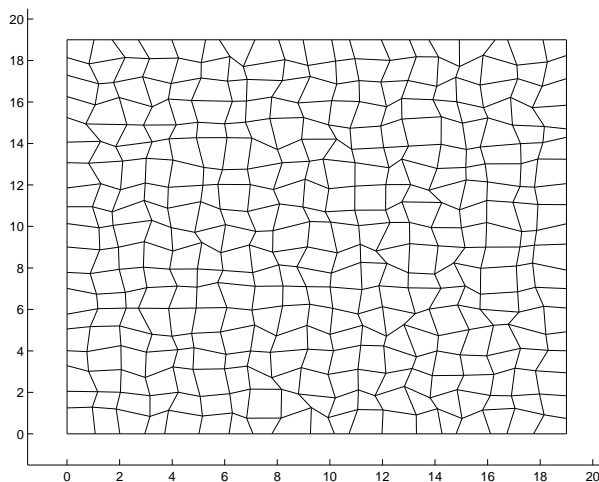


Figure K.1: An example of the type of grid we will generate.

$n_x \times n_y$ gridcell square grid. For a gridpoint in the interior, $\mathbf{x}_i = (x_i, y_i)$, we generate a perturbed gridpoint $\tilde{\mathbf{x}}_i = (x_i + R_x, y_i + R_y)$, where R_x and R_y satisfies $-r_x/2 < R_x < r_x/2$, $-r_y/2 < R_y < r_y/2$ and r_x and r_y are a given a set of pertubation coefficients.

After all point in the grid have been perturbed, we check if any gridcell has become non-convex. For g , all cells is convex, as they should be.

Appendix L

Fluid particle

In this Appendix we give a more detailed description of the concept of a fluid particle and its velocity.

Before we discuss the fluid particle, we must define the representative elementary volume [7].

L.1 Representative Elementary Volume

The porosity of a porous material is the fraction of the bulk volume of the material occupied by voids. We define the Representative Elementary Volume (REV) through the concept of porosity. Let P be a mathematical point inside the domain occupied by the porous medium. Consider a volume ΔU_i much larger than a single pore, for which P is the centroid. For this volume we determine the ratio:

$$\phi_i \equiv \phi_i(\Delta U_i) = (\Delta U_v)_i / \Delta U_i \quad (\text{L.1})$$

where $(\Delta U_v)_i$ is the volume of void space within ΔU_i . Repeating the same procedure, a sequence of values $\phi_i(\Delta U_i), i = 1, 2, 3, \dots$ is obtained by gradually shrinking the size of ΔU_i around P as a centroid: $\Delta U_1 > \Delta U_2 > \Delta U_3 \dots$. For large values of ΔU_i , the ratio ϕ_i may undergo gradual changes as ΔU_i is reduced, especially when the considered domain is inhomogeneous. Below a certain value of ΔU_i , depending on the distance of P from boundaries of inhomogeneity, these changes tend to decay, leaving only small-amplitude fluctuations that are due to the random distribution of pore sizes in the neighbourhood of P . However below a certain value ΔU_0 we suddenly observe large fluctuations in the ratio ϕ_i . This happens as the dimensions of ΔU_i approach those of a single pore. Finally, as $\Delta U_i \rightarrow 0$, converging on the mathematical point P , ϕ_i will become either one or zero, depending on whether P is inside a pore or inside the solid matrix of the medium.

The medium's volumetric porosity $\phi(P)$ at the point P is defined as the

limit of the ratio ϕ_i as $\Delta U_i \rightarrow \Delta U_0$:

$$\phi(P) = \lim_{\Delta U_i \rightarrow \Delta U_0} \phi_i \{ \Delta U_i(P) \} = \lim_{\Delta U_i \rightarrow \Delta U_0} \frac{(\Delta U_v)_i(P)}{\Delta U_i} \quad (\text{L.2})$$

For values of $\Delta U_i < \Delta U_0$, we must consider the actual presence of pores and solid particles; in this range there is no single value that can represent the porosity at P. The volume ΔU_0 is therefore the representative elementary volume (REV).

Assuming that both ΔU_0 and ΔU_v vary smoothly in the vicinity of P, we have

$$\phi(P) = \lim_{P' \rightarrow P} \phi(P') \quad (\text{L.3})$$

which means that ϕ is a continuous function of the position of P within the porous medium.

Thus by introducing the concept of porosity and the definition of REV, we have replaced the medium by a fictitious continuum in which we may assign values of any property to any mathematical point in it.

L.2 Fluid particle and velocity

We will assume that both the fluid and the porous matrix are considered as continua filling the entire space. A *fluid particle* is defined as an ensemble of molecules included in a certain volume. This volume is associated with the REV of a fluid continuum.

The *velocity* of the fluid particle is the statistical average of the velocities of the individual molecules in the fluid particle.

Appendix M

Area of quadrilateral in 2D

The Prévost-method uses flux mapping from \mathcal{P} to \mathcal{R} that is based on a simple scaling of the fluxes by the Jacobi-determinant, J_m , of the midpoint of the cell.

In this Appendix we will calculate the area, A , of an irregular quadrilateral, Q . This will show that $A = J_m$.

Let $Q(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ be given as in Definition 5.1. We begin by splitting Q up into two triangles, T_1 and T_2 by joining \mathbf{x}_1 and \mathbf{x}_3 with a line. See Figure M.1. Let the corners of Q be $\mathbf{x}_1 = (x_1, y_1)$, $\mathbf{x}_2 = (x_2, y_2)$, $\mathbf{x}_3 =$

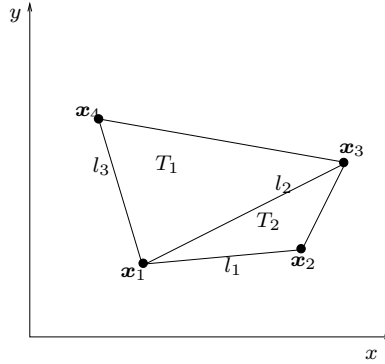


Figure M.1: Subtriangles T_1 and T_2 .

(x_2, y_2) and $\mathbf{x}_4 = (x_4, y_4)$. Define

$$l_1 = \{\text{Line joining } \mathbf{x}_1 \text{ and } \mathbf{x}_2\} \quad (\text{M.1})$$

$$l_2 = \{\text{Line joining } \mathbf{x}_1 \text{ and } \mathbf{x}_3\} \quad (\text{M.2})$$

$$l_3 = \{\text{Line joining } \mathbf{x}_1 \text{ and } \mathbf{x}_4\} \quad (\text{M.3})$$

$$\theta_i = \{\text{Angle between } l_i \text{ and positive } x\text{-axes, } i = 1, 2, 3.\} \quad (\text{M.4})$$

$$r_i = \{\text{Length of } l_i, i = 1, 2, 3.\} \quad (\text{M.5})$$

Let the point \mathbf{x}_1 be fixed then, for a convex Q , we have one of two cases, according to Definition 5.1. See Figure M.2.

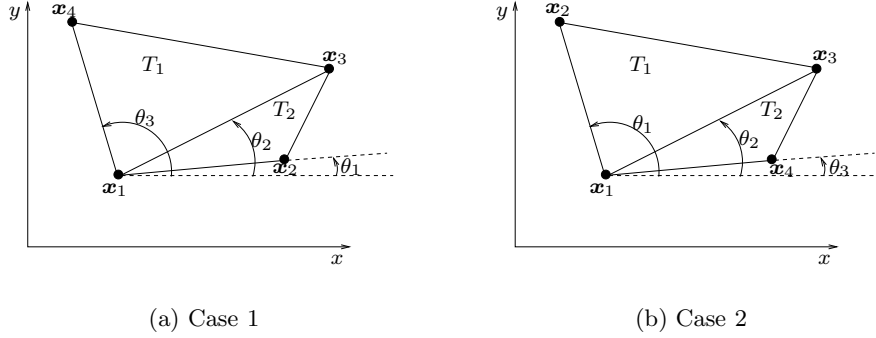


Figure M.2: Two possible ordering of the cornerpoints when \mathbf{x}_1 is fixed.

Define

$$K_1 = \frac{1}{2}r_2r_3 \sin(\theta_3 - \theta_2) \quad (\text{M.6})$$

$$K_2 = \frac{1}{2}r_1r_2 \sin(\theta_2 - \theta_1) \quad (\text{M.7})$$

Now consider the problem of calculating the areas A_1 and A_2 of the triangles T_1 and T_2 in case 1 in Figure M.2(a). From basic trigonometry we know that the areas are given by

$$A_1 = \frac{1}{2}r_2r_3 \sin(\theta_3 - \theta_2) = K_1 \quad (\text{M.8})$$

$$A_2 = \frac{1}{2}r_1r_2 \sin(\theta_2 - \theta_1) = K_2 \quad (\text{M.9})$$

For case 2 in Figure M.2(b) we get

$$A_1 = \frac{1}{2}r_1r_2 \sin(\theta_1 - \theta_2) = -\frac{1}{2}r_1r_2 \sin(\theta_2 - \theta_1) = -K_2 \quad (\text{M.10})$$

$$A_2 = \frac{1}{2}r_2r_3 \sin(\theta_2 - \theta_3) = -\frac{1}{2}r_2r_3 \sin(\theta_3 - \theta_2) = -K_1 \quad (\text{M.11})$$

So the area of Q is in either case given by

$$A = |K_1 + K_2| \quad (\text{M.12})$$

By expanding $\sin(\theta_3 - \theta_2)$ and $\sin(\theta_2 - \theta_1)$ this can be written

$$A = \frac{1}{2} |r_2 \cos(\theta_2)r_3 \sin(\theta_3) - r_2 \sin(\theta_2)r_3 \cos(\theta_3) + r_1 \cos(\theta_1)r_2 \sin(\theta_2) - r_1 \sin(\theta_1)r_2 \cos(\theta_2)| \quad (\text{M.13})$$

We want to express the area in terms of the coordinates of the corners of Q . To this we have

$$x_2 - x_1 = r_1 \cos(\theta_1) \quad (\text{M.14})$$

$$y_2 - y_1 = r_1 \sin(\theta_1) \quad (\text{M.15})$$

$$x_3 - x_1 = r_2 \cos(\theta_2) \quad (\text{M.16})$$

$$y_3 - y_1 = r_2 \sin(\theta_2) \quad (\text{M.17})$$

$$x_4 - x_1 = r_3 \cos(\theta_3) \quad (\text{M.18})$$

$$y_4 - y_1 = r_3 \sin(\theta_3) \quad (\text{M.19})$$

Inserting this in (M.13) we have

$$\begin{aligned} A = \frac{1}{2} & |(x_3 - x_1)(y_4 - y_1) - (y_3 - y_1)(x_4 - x_1) + \\ & (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)| = \\ & \frac{1}{2} |x_1 y_2 - y_1 x_2 + x_2 y_3 - y_2 x_3 + y_1 x_4 - x_1 y_4 + x_3 y_4 - y_3 x_4| \quad (\text{M.20}) \end{aligned}$$

We will not considered non-convex quadrilaterals in this work, but (M.20) is still valid for such Q .

Appendix N

C++ Implementation

C++ is an enhanced version of the C language. C++ includes everything that is part of C and adds support for object-oriented programming [28].

Object-oriented programming is a powerful way to approach the task of programming.

The elimination of spaghetti code became feasible with the invention of structured programming languages in the 1960s. These languages include Algol and Pascal. In loose terms, C is a structured language. Structured programming relies on well-defined control structures, code blocks, the absence of the GOTO, and stand-alone subroutines that support recursion and local variables. The essence of structured programming is the reduction of a program into its constituent elements. Using structured programming the average programmer can create and maintain programs that are up to 50,000 lines long.

Although structured programming has yielded excellent results when applied to moderately complex programs, even it fails at some point, after a program reaches a certain size. To allow more complex programs to be written, a new approach to the job of programming was needed. Towards this end, object-oriented programming was invented. OOP takes the best of the ideas embodied in structured programming and combines them with new concepts that allow you to organize your programs more effectively. Object-oriented programming encourages you to decompose a problem into its constituent parts. Each component becomes a self-contained object that contains its own instructions and data that relate to that object. In this way, complexity is reduced and the programmer can manage larger programs.

All OOP languages, including C++, share three common defining traits: encapsulation, polymorphism, and inheritance.

N.1 Source

N.1.1 Point inside quadrilateral

The source code for the point-inside-quadrilateral problem is given in Figure N.1.

```

bool inside(double xi, double yi, double x1, double x2, double x3, double x4,
            double y1, double y2, double y3, double y4) {
    int l;

    if (checkpoint(xi, yi, x1, y4, y1, y2, y3, &l)) return true;
    if (f=checkpoint(xi, yi, x2, y1, y2, y3, y4, &l)) return true;
    if (f=checkpoint(xi, yi, x3, y2, y3, y4, y1, &l)) return true;
    if (f=checkpoint(xi, yi, x4, y3, y4, y1, y2, &l)) return true;
    if (f=checkedge(xi, yi, x1, y1, x2, y2, &l)) return true;
    if (f=checkedge(xi, yi, x2, y2, x3, y3, &l)) return true;
    if (f=checkedge(xi, yi, x3, y3, x4, y4, &l)) return true;
    if (f=checkedge(xi, yi, x4, y4, x1, y1, &l)) return true;
    if (l==1 || l==3) return true;
    return false;
}

bool checkpoint(double x, double y, double x2, double y1,
                double y2, double y3, double y4, int *l) {
    if (y==y2) {
        if (x==x2) return true;
        else if (x>x2) {
            if (y==y1) {
                if (y>y4 && y<y3) *l=*l+1;
                else if (y<y4 && y>y3) *l=*l+1;
            }
            else if (y==y3) {
                if (y>y1 && y<y4) *l=*l+1;
                else if (y<y1 && y>y4) *l=*l+1;
            }
            else if (y>y1 && y<y3) *l=*l+1;
            else if (y<y1 && y>y3) *l=*l+1;
        }
    }
    return false;
}

bool checkedge(double x, double y, double x1, double y1, double x2,
                double y2, int *l) {
    if (y==y1 && y1==y2) {
        if ((x>x1 && x<x2) || (x>x2 && x<x1)) return true;
    }
    else if ((y>y1 && y<y2) || (y>y2 && y<y1)) {
        if (x1==x2) {
            if (x > x1) *l=*l+1;
            else if (x==x1) return true;
        }
        else {
            double a=(y2-y1)/(x2-x1);
            double xp=(y-y1)/a+x1;
            if (x>xp) *l=*l+1;
            else if (x==xp) return true;
        }
    }
    return false;
}

```

Figure N.1: C++ source

Appendix O

Matlab visualization

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in a familiar mathematical notation [12].

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN.

O.1 Cell arrays

Cell arrays can be used to represent streamlines in MATLAB. Cell arrays are MATLAB arrays whose elements are cells. Each cell in a cell array can hold any MATLAB data type, including numeric arrays, text, symbolic objects, cell arrays and structures. For example, one cell of a cell array might contain a numeric array, another an array of text strings, and another a vector of complex values.

O.2 Plotting streamlines

A simplified version of the streamline plotting routine is given in Figure O.1.

```

function A=plotSL( gf , sf )
    A=readData( sf );
    clf;
    plotgrid( gf );
    figure( gcf );
    hold on;
    numLines=length(A);
    for j=1:numLines
        x1=A{ j }( 1 ,:);
        x2=A{ j }( 2 ,:);
        x3=A{ j }( 3 ,:);
        x4=A{ j }( 4 ,:);
        y1=A{ j }( 5 ,:);
        y2=A{ j }( 6 ,:);
        y3=A{ j }( 7 ,:);
        y4=A{ j }( 8 ,:);
        u1=A{ j }( 9 ,:);
        u2=A{ j }( 10 ,:);
        v1=A{ j }( 11 ,:);
        v2=A{ j }( 12 ,:);
        Ax=A{ j }( 13 ,:);
        Ay=A{ j }( 14 ,:);
        xs=A{ j }( 15 ,:);
        ys=A{ j }( 16 ,:);
        xe=A{ j }( 17 ,:);
        ye=A{ j }( 18 ,:);
        te=A{ j }( 19 ,:);
        N=length(te);
        MAX_T=1000; % upper limit
        for i=1:N
            if te(i)>MAX_T, te(i)=MAX_T; end;
            t=linspace( 0 , te ( i ) , 100 );
            if (Ax(i)==0)
                xi=u1(i)*t+xs(i);
            else
                if u1(i)~=0
                    xi=(1/Ax(i))*((Ax(i)*xs(i)+u1(i))*exp(Ax(i)*t)-u1(i));
                else
                    xi(length(t):-1:1)=xe(i)*exp(-u2(i)*t);
                end
            end
            if (Ay(i)==0)
                eta=v1(i)*t+ys(i);
            else
                if v1(i)~=0
                    eta=(1/Ay(i))*((Ay(i)*ys(i)+v1(i))*exp(Ay(i)*t)-v1(i));
                else
                    eta(length(t):-1:1)=ye(i)*exp(-v2(i)*t);
                end
            end;
            [x,y]=iso( xi , eta , x1(i) , y1(i) , x2(i) , y2(i) , x3(i) , y3(i) , x4(i) , y4(i) );
            plot(x,y);
        end;
    end;
    figure( gcf );
return;

```

Figure O.1: Plotting routine

Bibliography

- [1] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geosciences*, 6:405–432, 2002.
- [2] R.A. Adams. *Calculus: A complete course*. Addison-Wesley, third edition, 1995.
- [3] A.K. Ask. Simulering av to-fase flyt ved hjelp av operatorsplitting langs strømmlinjer. Master’s thesis, University of Bergen, Dept. of Mathematics, 2000.
- [4] K. Aziz. Reservoir simulation grids. opportunities and problems. *Journal of Petroleum Technology*, (July):658–663, 1993.
- [5] R.P. Batycky. *A three-dimensional two-phase field scale streamline simulator*. PhD thesis, Stanford University, Dept. of Petroleum Engineering, 1997.
- [6] R.P. Batycky, M.J. Blunt, and M.R. Thiele. A 3D field-scale streamline-based reservoir simulator. *SPE Reservoir Engineering*, (Nov.):246–254, 1997.
- [7] J. Bear. *Dynamics of fluids in porous media*. American Elsevier Publishing Company, New York, 1972.
- [8] P.G. Ciarlet. *Finite element method for elliptic problems*. North Holland, Amsterdam, 1978.
- [9] A. Datta-Gupta and M.J. King. A semianalytical approach to tracer flow modeling in heterogeneous permeable media. *Advances in Water Resources*, 18(9), 1995.
- [10] L.E. Engevik. *Forelesninger i kontinuumsmekanikk*. University of Bergen, Department of mathematics, 1980.
- [11] D.J. Goode. Particle velocity interpolation in block-centered finite difference groundwater flow models. *Water Resources Research*, 26(5):925–940, 1990.

- [12] D. Hanselman and B. Littlefield. *MATLAB The language of technical computing*. Prentice Hall, 1997.
- [13] Torbjørn Ole Widnes Johansen. Karakteristikk baserte område dekomponeringsteknikkar for simulering av eit grunnvassreservoar i to rommelege dimensjonar. Master's thesis, University of Bergen, Dept. of applied mathematics, 1996.
- [14] J. Kevorkian. *Partial differential equations : analytical solution techniques*. Chapman & Hall, New York, 1993.
- [15] M.J. King and A. Datta-Gupta. Streamline simulation: A current perspective. *In Situ*, 22(1):91–140, 1998.
- [16] W. L. Lake. *Enhanced Oil Recovery*. Prentice Hall, New York, first edition, 1989.
- [17] N. Lu. A semianalytical method of path line computation for transient finite-difference groundwater flow models. *Water Resources Research*, 30(8):2449–2459, 1994.
- [18] M. Muskat. *Physical principles of oil production*. International Human Resources Development Corporation, Boston, 1981.
- [19] R.L. Naff, T.F. Russell, and J.D. Wilson. Shape functions for velocity interpolation in general hexahedral cells. *Computational Geosciences*, 2002.
- [20] J.M. Nordbotten and G.T. Eigestad. Monotonicity conditions for control volume methods on general grids; application to mpfa. In *Proceedings of the 17th Nordic Seminar on Computational Mechanics.*, 2003.
- [21] D. W. Peaceman. *Fundamentals of numerical reservoir simulation*. Elsevier scientific publishing company, 1977.
- [22] D.W. Pollock. Semi-analytical computation of path lines for finite-difference models. *Ground Water*, 26(6):743–750, 1988.
- [23] D.K Ponting. Corner point geometry in reservoir simulation.
- [24] M. Prévost, M. G. Edwards, and M.J. Blunt. Streamline tracing on curvilinear structured and unstructured grids. *SPE Journal*, (June):139–148, 2002.
- [25] P.A. Raviart and J.M. Thomas. A mixed finite element method for 2nd order elliptic problems. *Lecture Notes in Mathematics 606*, pages 292–315, 1977.

- [26] 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, Calgary, Canada, 2000*.
- [27] A.L. Schafer-Perini and J.L. Wilson. Efficient and accurate front tracking for two-dimensional groundwater flow models. *Water Resources Research*, 27(7):1471–1485, 1991.
- [28] H. Schildt. *Teach yourself c++*. Osborne McGraw-Hill, New York, third edition, 1998.
- [29] J.M. Shafer. Reverse pathline calculation of time-related capture zones in nonuniform flow. *Ground Water*, 25(3):283–289, 1987.
- [30] M.R. Thiele. Streamline simulation. In *Proceedings of the Sixth International Forum on Reservoir Simulation, Fuschl, Austria, 2001*.
- [31] S.K. Verma. *Grids for reservoir simulation*. PhD thesis, Stanford University, Department of Petroleum Engineering, 1996.