Flexible finite volumes for tracer transport in coastal regions

S. van der Baan, P. Wilders

Abstract

This paper presents a method for the computation of transport of contaminants in surface waters on unstructured hybrid grids. A positive space discretization scheme is used for the advective flux, for time integration we use an implicit method. Hybridization is done by generalization of a solver for triangular grids. Numerical experiments are done for both triangular and quadrilateral grids for a model, containing a strip along the Dutch coast.

1 Introduction

The computation of transport of contaminants in surface waters is an important task in environmental engineering. Measurements of concentrations are difficult to obtain because of the lack of eligible tracers or the environmental harmfulness of these. Therefore numerical computations are important. These numerical computations are done in a wide variety of spatial domains with complex geometry and on time scales of several months or more. These circumstances make it necessary to have the disposal of flexible simulation tools.

One step in this direction is the use of implicit methods for time-integration, see [10]. This makes it possible to use large time steps that are required by the large time scales of the simulation. A number of methods have been introduced in the past that make it possible to represent the considered physical domain better by a computational domain. Domain decomposition has been introduced to model domains that consist of several parts that differ in shape. Curvilinear grids have been used to better fit the domain boundary. Even more flexibility is provided by using unstructured grids. Usually these unstructured grids consist of either quadrilaterals or (more often) of triangles. In this paper we enhance the flexibility by considering hybrid grids, i.e., grids of which the cells can have a varying number of edges.

This kind of hybridization has already been applied in case of structured grids in aerodynamic computations. Here grids are often highly stretched in near-wall regions for reasons of numerical accuracy. It appears that the use of quadrilaterals in these regions enhances the accuracy. At the same time it is more efficient because less cells have to be used. A
combination of quadrilaterals in near-wall regions and triangles elsewhere appears to be the best compromise between accuracy and efficiency [9].

In this paper, the strategy for the transport computation will be basically the same as the one proposed in [10]. An Eulerian depth-averaged model is considered, the transport equation is given in the so-called 2DH formulation. The coefficients (water depth and velocity field) are considered to be given by input data. We use implicit time stepping to enhance the robustness of the method. For the space discretization, a finite volume approach on an unstructured grid is adopted. In [10] only triangular grids have been considered. Here, we extend this method by using hybrid grids. The theory will be given using general hybrid grids, the numerical experiments are done on a grid consisting of quadrilaterals.

We apply this method to the Kuststrookmodel, a strip of $60 \times 400$ km in the North Sea along the Dutch coast. Residual currents in the domain are in north-east direction which means that contaminants that are transported in the river Rhine float into sea at Hoek van Holland and drift to the Waddenzee, a protected area in the Northern Netherlands. We are interested in the time scales of this process.

The outline of this paper is as follows: In Section 2, the physical model is given. The global design of the solver of Wilders for triangular meshes is described in Section 3. In Section 4 the changes with respect to hybridization of the method are explained. Some results of numerical experiments are given in Section 5. We end with some final remarks in Section 6.

2 The model equations

The physical model consists of an advection-diffusion equation that models the transport of the constituent concentration combined with flow equations that describe the advective velocities of the constituent. In case of tracer transport, flow and transport are often modeled separately because of the much larger time scale of the transport equation compared to the flow equations. Transport is then computed with an a priori known velocity field for the whole domain and the whole simulation period.

The tracer concentration is described by the following transport equation in $\mathbb{R}^3 \times \mathbb{R}$:

$$\frac{\partial c}{\partial t} + \nabla \cdot (uc - D\nabla c) = 0,$$

in which $c(x,t)$ is the concentration of the tracer, $u(x,t)$ is the flow velocity and $D(u)$ is a dispersion tensor. Of course, $c \geq 0$ everywhere.

The model that we use is simplified in a few ways:

1. the so-called 2DH approach is applied, i.e.,
   
   (a) $x \in \mathbb{R}^2$,
   
   (b) $u = Hv$ with $v(x,t)$ the depth-averaged flow velocity and $H(x,t) > 0$ the total water depth, i.e. the distance between bottom area and water surface,
(c) $c$ becomes a depth-averaged concentration, $\frac{\partial c}{\partial t}$ is replaced by $\frac{\partial Hc}{\partial t}$,
(d) $D = HD_H$ with $D_H$ the depth-averaged dispersion tensor.

2. $H$ is time-independent within a time step,

Using these adjustments, the transport equation becomes

$$\frac{\partial Hc}{\partial t} + \nabla \cdot (uc - D\nabla c) = 0. \quad (1)$$

The variables $u$ and $H$ are computed by an external flow solver and are considered to be given for the transport computation. Furthermore, we assume conservation of mass, which, because of 2., reduces to $\nabla \cdot u = 0$. In our numerical experiments the given flow does not satisfy this property. We therefore adjust the velocity field in a preprocessing step that is described in Section 3.5.

The boundary conditions are

$$c(x,t) = 0, \quad x \in \Gamma_{in},$$
$$D\nabla c \cdot n = 0, \quad x \in \Gamma \setminus \Gamma_{in}.$$ 

Here, $n$ denotes the outward unit normal on the boundary $\Gamma$ and $\Gamma_{in} = \{x \in \Gamma : u \cdot n < 0\}$. This implies that on closed boundaries where $u \cdot n = 0$, the applied boundary condition $D\nabla c \cdot n = 0$ is equivalent with a prescription of the total flux $(-uc + D\nabla c) \cdot n$.

2.1 Comments on physical model and numerical implications

Coupling of flow and transport computation In a continuous sense, the flow is independent of the concentration in the case of tracer flow. Therefore, flow and transport can be computed independently, i.e., the concentration can be computed with a known velocity field that is computed beforehand.

In a discrete sense, one can still apply a certain coupling between the computation of flow and concentration by using the same grid and discretization methods in time and space. Instead, we use externally computed velocity fields, and apply different time and space discretization schemes for the computation of the concentration.

Conservation of mass For the construction of the space discretization scheme, we assume conservation of mass. This assumption is only met in general if the flow and concentration are computed in the discrete coupled sense of the previous comment. In our situation, conservation of mass in a discrete sense is not satisfied. The maximum value (over all grid cells) of $\nabla \cdot u$ is of the order of $10^{-3}$. As a consequence, the properties of the derived numerical scheme do not strictly hold and we may expect instability because of the introduction of unmeant source terms. In order to deal with these instabilities, a preprocessing step of the velocity field is introduced.

Boundary conditions The boundary conditions for open boundaries are a subject of consideration. An alternative for inflow boundaries might be to use concentration information at previous time levels and to take the tidal variation into account.
# 3 Triangular solution method

The transport solver that is described here is an extension of an existing transport solver on triangular meshes. Details of that solver can be found in \[10, 12\]. We will give an overview here. The extension with respect to hybridization is described in Section 4.

## 3.1 Space discretization

In \[10\] a finite volume approach is used for space discretization. The grid is considered to be composed of triangles. The finite volumes are the triangular cells of the grid (cell-centered approach). The cells are denoted by \(\Delta\), the edges of the cells are the connections between the grid points and are denoted by \(e\). Sometimes the subscript \(\Delta\) is added to indicate that edge \(e_\Delta\) belongs to \(\Delta\). For convenience we write the transport equation (1) in the following form

\[
\frac{\partial H_c}{\partial t} + \nabla \cdot f - \nabla \cdot g = 0, \quad (2)
\]

in which \(f = uc\) and \(g = D\nabla c\). We integrate (2) over some cell \(\Delta\) and apply Green’s divergence theorem to approximate (2) by

\[
|\Delta| \frac{dH_c}{dt} + \sum_{e \in \partial \Delta} \left( \int_{e} f \cdot n_{e,\Delta} d\sigma - \int_{e} g \cdot n_{e,\Delta} d\sigma \right) = 0, \quad (3)
\]

where summation is carried out over all cell edges \(e\) of the cell, \(n_{e,\Delta}\) denotes the outward unit normal on the edge \(e\) (outward from the perspective of cell \(\Delta\)), \(|\Delta|\) is the area of \(\Delta\) and \(H\) and \(c\) are now constant functions over the cell. The edge integrated fluxes \(\int_{e} f \cdot n_{e,\Delta} d\sigma\) and \(\int_{e} g \cdot n_{e,\Delta} d\sigma\) will be approximated by respectively \(f_{e,\Delta}|e|\) and \(g_{e,\Delta}|e|\), in which \(f_{e,\Delta}\) and \(g_{e,\Delta}\) are the advective and the viscous numerical flux functions and \(|e|\) is the length of \(e\). The functions \(f_{e,\Delta}\) and \(g_{e,\Delta}\) can be regarded as averages over the edge \(e\) of the advective and viscous flux respectively.

The numerical flux functions will be expressed in terms of the geometry of a 6-point molecule around edge \(e\), see Figure 1. The cells 1 and 2 are the so-called primary cells of edge \(e\) and the cells 3, ..., 6 are the secondary cells. The discretization of both the advective and the viscous flux is based on this 6-point molecule.

## 3.2 Advective flux

In this section, we derive an expression for \(f_{e,\Delta}\) that implies some desired properties to be mentioned shortly. In this derivation we neglect the diffusion, i.e.,

\[
\int_{e} g \cdot n_{e,\Delta} d\sigma = 0. \quad (4)
\]

Furthermore, in Section 3.5 we describe a correction of the velocity field that is introduced to ascertain mass-conservation of the flow. Substitution of \(u\) by this corrected velocity...
field, making the assumption of time-independent $H$ within a time step (assumption 2 in Section 2) together with (4) gives that (3) is consistent with

$$|\Delta|H \frac{dc}{dt} + \sum_{e \in \partial \Delta} f_{e, \Delta}(c, u)|e| = 0$$

for cell $\Delta$. Taking together all cells, we get the system of equations

$$M \frac{dc}{dt} = -F(c, u). \quad (5)$$

The matrix $M$ is a diagonal matrix, containing the areas and the water depths of the cells. The function $F$ depends on the actual discretization scheme for the advective flux. Some desirable properties of a numerical scheme can be obtained by considering a so-called advective K-approximation. $F(c, u)$ is such an approximation if there exists a matrix function $\tilde{Q} = (\tilde{q}_{jk})$, $\tilde{q}_{jk} = \bar{q}_{jk}(c, u)$ with

$$\tilde{q}_{jk} \leq 0, \quad j \neq k \quad \text{and} \quad \sum_k \tilde{q}_{jk} = 0, \quad \text{for all } c, \quad (6)$$

such that

$$F_j(c, u) = \sum_k \tilde{q}_{jk}c_k \quad \text{for all } j \text{ in the interior.}$$

An advective K-approximation implies among others that local minima are nondecreasing and local maxima are non-increasing [4] and that the solution is nonnegative if the input is nonnegative [3, 5].

We must define the function $f_{e, \Delta}$ in such a way that an advective K-approximation results. For $f_{e, \Delta}$ we take the Enquist-Osher function [2] which for a single linear flux is of the form

$$f_{e, \Delta}(c_e^L, c_e^R) = \frac{u_{n_e, \Delta} + |u_{n_e, \Delta}|c_e^L}{2} + \frac{u_{n_e, \Delta} - |u_{n_e, \Delta}|c_e^R}{2}. \quad (7)$$

---

Figure 1: *Edge molecule for the discretization of the advective and viscous flux on $e$.***
Here, \(c^L_e\) and \(c^R_e\) are estimated values of \(c\) on edge \(e\) and \(u_{n.e.\Delta} = u_e \cdot n_{e.\Delta}\) with \(u_e\) the edge value of the velocity \(u\).

We will now give the requirements for \(c^L_e\) such that (7) is an advective K-approximation. Remark that, by applying a renumbering of cells, this is equivalent to giving the requirements for \(c^R_e\). Since we assumed a divergence-free velocity field, i.e., \(\sum_{e \in \partial \Delta} f_{e.\Delta}(c_1, c_1) = 0\), we can write

\[
\sum_{e \in \partial \Delta} f_{e.\Delta}(c^L_e, c^R_e) = \sum_{e \in \partial \Delta} \{f_{e.\Delta}(c^L_e, c_1) - f_{e.\Delta}(c_1, c_1)\} + \sum_{e \in \partial \Delta} \{f_{e.\Delta}(c^L_e, c^R_e) - f_{e.\Delta}(c^L_e, c_1)\}.
\]

By (7), it follows that

\[
\sum_{e \in \partial \Delta} f_{e.\Delta}(c^L_e, c^R_e) = \sum_{e \in \partial \Delta} \alpha^L_{e.\Delta} + \sum_{e \in \partial \Delta} \alpha^R_{e.\Delta},
\]

with

\[
\alpha^L_{e.\Delta} = \begin{cases} u_{n.e.\Delta} \frac{c^L_e - c_1}{c_1 - c_k} (c_1 - c_k), & u_{n.e.\Delta} > 0, \quad k \in \{2, 3, 4\}, \\ 0, & \text{elsewhere}, \end{cases}
\]

\[
\alpha^R_{e.\Delta} = \begin{cases} -u_{n.e.\Delta} \frac{c^R_e - c_1}{c_2 - c_1} (c_1 - c_2), & u_{n.e.\Delta} < 0, \\ 0, & \text{elsewhere}. \end{cases}
\]

The second part of (6), \(\sum_k \tilde{g}_{jk} = 0\), is now satisfied. For the other requirement of (6), we must have that

\[
\frac{c^L_e - c_1}{c_1 - c_k} \geq 0 \quad \text{for at least one } k \in \{2, 3, 4\},
\]

and

\[
\frac{c^R_e - c_1}{c_2 - c_1} \geq 0.
\]

Remark that (9) is equivalent with

\[
\frac{c^L_e - c_1}{c_2 - c_1} \leq 1
\]

for all cells \(\Delta\) and all \(e \in \partial \Delta\). This can be deduced by writing (9) in terms of the local numbering of the cell at the other side of edge \(e\), i.e., the indices 1 and 2 change order and \(c^R_{e.\Delta_1} = c^L_{e.\Delta_2}\).

To explain how to reconstruct \(c^L_e\) such that it satisfies (8), (10), we use the following notation. Let \(P_a\) and \(P_b\) be two points in \(\mathbb{R}^2\). With \(t_{ab}\) we denote the vector from \(P_a\) to \(P_b\) and with \(n_{ab}\) the normal, pointing to the right on the segment between \(P_a\) and \(P_b\). The length of the vectors \(t_{ab}\) and \(n_{ab}\) is denoted by \(|t_{ab}|\) and is equal to the length of the segment \(P_aP_b\). The only exception is \(n_e\), the normal to edge \(e\), which we defined to have length 1. With \(t_{1e}\), we denote the vector pointing from cell center \(P_1\) to the midpoint \(P_m\) of the edge.
We assume that the mesh is such that
\[ n_{13} \cdot t_{1e} \geq 0 \quad \text{and} \quad n_{41} \cdot t_{1e} \geq 0, \]
i.e., point \( P_3 \) must be above the line through \( P_1 \) and \( P_m \) whereas \( P_4 \) must be below that line. This is called a TVD triangulation in [7].

Now, let \( \nabla_1 \) be an approximation of the gradient of \( c \) in cell center \( P_1 \) and let \( \psi \) be a scalar function of one variable. We take \( c^L_e \) of the following form:
\[ c^L_e = c_1 + \psi \delta_{e1}, \quad \delta_{e1} = \nabla_1 \cdot t_{1e}. \quad (11) \]

For the computation of \( \nabla_1 \), we apply Green-Gauss reconstruction on the triangle \( P_1 P_3 P_4 \), with area \( A_{134} = \frac{1}{2} |t_{13} \cdot n_{14}| \), i.e.,
\[ \nabla c|_{P_1} \approx \frac{1}{A_{134}} \int_{\Delta_{134}} \nabla c \, dx = \frac{1}{A_{134}} \int_{\partial \Delta_{134}} c n \, ds \approx \]
\[ \nabla_1 := \frac{1}{A_{134}} \sum_{e \in \partial \Delta_{134}} c_e n_e = \ldots = \frac{n_{41}(c_1 - c_3) + n_{13}(c_1 - c_4)}{A_{134}}. \]

Here, \( c_e \) is the average of the concentration values at the two endpoints of edge \( e \). The accuracy of \( \nabla_1 \) depends on the smallest angle in the triangle \( P_1 P_3 P_4 \). Highly stretched triangles may lead to inaccurate approximations. For the consequences of this on hybridization, see Section 4.

For the limiting function \( \psi \), we take \( \psi = \psi(r) \), with
\[ r = \frac{\delta_{21}}{\delta_{e1}} = \frac{c_2 - c_1}{\nabla_1 \cdot t_{1e}}. \quad (12) \]

A sufficient condition for \( \psi \) to imply (8) and (10) is
\[ \psi = 0, \quad r < 0, \]
\[ 0 \leq \psi \leq r, \quad r \geq 0 \quad (13) \]

since

1. if \( r < 0 \), then the requirements (8) and (10) follow immediately from (11),

2. if \( r \geq 0 \), then requirement (10) follows immediately from (11) and (12). For requirement (8) we note that
\[ \frac{c^L_e - c_1}{c_1 - c_k} = \psi \frac{(n_{41} \cdot t_{1e})(c_1 - c_3) + (n_{13} \cdot t_{1e})(c_1 - c_4)}{A_{134}(c_1 - c_k)}. \]

We can distinguish two cases:

(a) \((c_1 - c_3 < 0 \land c_1 - c_4 < 0) \lor (c_1 - c_3 > 0 \land c_1 - c_4 > 0)\): both \( k = 3 \) and \( k = 4 \) imply (8),
(b) \((c_1 - c_3 < 0 \land c_1 - c_4 > 0) \lor (c_1 - c_3 > 0 \land c_1 - c_4 < 0)\): either \(k = 3\) or \(k = 4\), in dependence of the ratio \(\frac{n_{13} \cdot \mathbf{t}_{13}}{n_{13} \cdot \mathbf{t}_{14}}\) implies (8).

An example of a \(\psi\) that satisfies (13), is the \(R-1\) limiter from Zijlema [13]:

\[
\psi = 2 \frac{(r + |r|)r}{(1 + r)^2}.
\]

A plot of this function is given in Figure 2.

![Figure 2: Graphical representation of the \(R-1\) limiter from Zijlema.](image)

**Boundary treatment** If the edge \(e\) belongs to \(\Gamma\), then we must use the boundary value information, the above procedure for determining \(c_e\) cannot be used. We can distinguish two situations: In case of a Dirichlet condition, we use the prescribed edge value of the concentration. In case of a zero flux condition the edge value is set equal to the concentration value of the corresponding cell center.

### 3.3 Viscous flux

For the viscous flux the gradient of \(c\) on edge \(e\) is needed. This gradient is approximated on the shadow volume \(P_1 P_2 P_3\), see Figure 1. In fact, the area weighted average of the gradients on the triangles \(P_1 P_2 P_3\) and \(P_1 P_2 P_4\) is taken. The concentration values in the points \(P_2\) and \(P_3\) are obtained by linear interpolation between \(P_4\), \(P_5\) and \(P_3\), \(P_6\) respectively. The gradients in both triangles are

\[
\nabla c|_{1a2} = \frac{n_{a1}(c_2 - c_1) + n_{12}(c_a - c_1)}{2A_{1a2}},
\]

\[
\nabla c|_{12b} = \frac{n_{1b}(c_2 - c_1) + n_{21}(c_b - c_1)}{2A_{12b}},
\]

in which \(c_a = (c_4 + c_5)/2\), \(c_b = (c_6 + c_3)/2\) and \(A_{1a2} = \frac{1}{2} t_{1a} \cdot \mathbf{n}_{12}\) and \(A_{12b} = \frac{1}{2} t_{1b} \cdot \mathbf{n}_{21}\) are the areas of the triangular shadow volumes. The resulting formula for \(g_{e,\Delta}\) is

\[
g_{e,\Delta} = \frac{(\mathbf{D} \cdot \mathbf{n}_e)(c_2 - c_1) + (\mathbf{D} \cdot \mathbf{n}_{12})(c_a - c_b)}{2A_{1a2}},
\]
in which the area $A_{1a2b} = \frac{1}{2}t_{12} \cdot n_{ab}$. So the full 6-point molecule is used for defining the viscous numerical flux function. This choice of discretization does not guarantee a viscous K-approximation (the equivalent of an advective K-approximation for the viscous terms). For the moment we leave this as it is. We assume that the contribution of the viscous term is much smaller than the contribution of the advective term.

Boundary treatment The edge $e$ itself or some of the four chosen neighboring edges can be on the boundary. In the latter case, we’ve left out the corresponding cell center in the construction of the triangular shadow volume. For example, if cell $P_4$ is absent, then $P_b$ coincides with $P_3$. If both $P_3$ and $P_4$ are absent, then only the other triangular shadow volume $P_1P_aP_2$ is taken into account. If the edge $e$ itself is on the boundary, then either the viscous flux is zero or there holds a Dirichlet condition. In the latter case we take the area-weighted combination of the shadow volumes $P_1P_1P_m$ and $P_1P_mP_3$, with $P_m$ the middle point of the boundary edge.

In corners, either $P_3$ or $P_4$ is absent as well. If the corresponding boundary edge has a Neumann condition, then the corresponding triangular shadow volume is left out. If there is a Dirichlet condition on that edge, then we replace $P_3$ and $P_4$ by the middle point of the corresponding neighboring boundary edge and take its prescribed edge value.

3.4 Time integration

Transport phenomena have long time scales. An implicit time stepping method is used to be able to use large time steps. If we apply the same reasoning as in the derivation of (5), i.e. assuming time-independent $H$ within a time step and correction of the velocity field, but this time including the viscous flux, we get

$$M^d c d \tau = P \imath.$$  

The matrix $M$ is a diagonal matrix as explained in Section 3.2 containing the cell values of the water depth multiplied by the area of the cell. The matrix $P$ is a sparse matrix (maximal 10 nonzero elements per row) with entries depending on $c$, containing the contribution of the advective and viscous flux. For time integration we use the linearly implicit trapezoidal rule:

$$\left( \frac{M^n}{\tau_n} - \frac{1}{2} \left( \frac{\partial P \imath}{\partial c} \right)^n \right) (c^{n+1} - c^n) = (P \imath)^n.$$  

Here, $\tau_n$ denotes the time step. The linear system (14) is solved iteratively with ILU-preconditioned BiCGSTAB.

3.5 Treatment of time-dependent coefficients

The time-dependent coefficients $u$ and $H$ are needed for the whole transport simulation period. This implies a very large data storage. A great reduction of storage required can be
accomplished by making use of the periodicity of the coefficients. Often only the field data for a few periods are stored and these are periodically extrapolated to the whole simulation period.

There are some aspects in this procedure that must be looked after. The residual velocities that we are interested in are some orders of magnitude smaller than the tidal velocities that are input for the transport solver. A small difference in the actual tidal period and the length of the interval for which the field data are given can have a large effect on the residual velocities. Also the length of the tidal period depends on the position in the domain and on the tidal phase (ranging from neap-tide to spring-tide). Tidal periods in our domain of interest vary between 12h18min and 12h45min, with an average of 12h25min, see [8]. To include these tidal differences, thereby reducing errors in residual velocities, the time-dependent coefficients should be tabulated for the duration of a half moon cycle, which corresponds to some 28 or 29 tidal periods. Although this may be a considerable reduction in data storage compared to the data that are needed for the whole simulation period, the amount of data for 28 periods is still very large. In the numerical experiments in Section 5 we have used velocity fields that are tabulated for the duration of one tidal period.

Mass conservation ($\nabla \cdot u = 0$ in a specific discrete sense) is needed for a stable transport computation (avoidance of mass sources and sinks) and for obtaining a positive discretization scheme for the advective flux. Our input velocity fields do not satisfy this property. The velocity fields are changed in a preprocessing step in order to restore this property. Two approaches are possible:

1. The whole velocity field can be set to the (unique) velocity field that is divergence free and is most similar (in some norm) to the original velocity field.

2. Changing the velocity field on a local basis, i.e. per cell adding a small term to every edge, so that it becomes divergence free. In formula:

$$\ddot{u}_{ne,\Delta} = u_{ne,\Delta} - \sum_{e \in \partial \Delta} \frac{u_{ne,\Delta}^2}{e_{e,\Delta}} \sum_{e \in \partial \Delta} \dot{u}_{ne,\Delta}.$$  

Consequently, the velocity field is not unique per edge anymore and conservation of mass for transport is dropped. See also [11].

Option 1 is computationally more expensive. We use the second option here.

3.6 Edge-based data structure

The design of a data structure is related to a number of facts. The unknown concentrations are defined in the cell centers, i.e., there is a one-to-one correspondence between the unknowns and the grid cells. A cell-based data structure for the construction of the matrix-vector system (14) seems most suited from this point of view. However, the finite
volume approach that is used for the space discretization has a clear edge-based nature. Fluxes are needed at the edges of the cells and therefore values of all the variables are needed at the edges. Furthermore, this edge-based approach is conceptually simplest if the fluxes are computed only once and assigned to the two adjacent cells directly. The whole task can be done then with one loop over all edges.

A number of array-variables is needed to relate the edges to the 6-point molecule for that edge (see Section 3.1). One of them (plist1) gives for every edge the left and right adjacent cell number and the two vertices that are connected by the edge. An other array, plist2, gives for every cell the number of the three (in case of a triangular grid) adjacent cells. In the hybrid case that is described in the next section, the number of edges per cell is variable. This complicates the design of the data structure since the amount of positions in the array needed per cell/edge differs also, like for example in plist2. We solve this by using so-called offset arrays that give the starting positions for every cell or edge within the array.

4 Hybridization

In case of hybrid meshes we may encounter a $N$-point molecule as for example in Figure 3. The edge under consideration is denoted by $e$. Cells 1 and 2 are the cells left and right of edge $e$ respectively. $N_i$ is the number of edges of cell $i$. We want to determine the advective and the viscous flux on edge $e$ thereby using a generalization of the advective and viscous flux discretization schemes of the triangular grid case. Two approaches are possible:

1. Design numerical flux functions that use, apart from the primary cells, all secondary cells $3, \ldots, N_1 + 1$ and $N_1 + 2, \ldots, N_1 + N_2$.
2. Select two secondary cells on both sides of the edge under consideration. Use these cells for the numerical flux functions of the triangular case as described in Section 3.

The second option is simpler to implement, because it is directly based on the existing triangular discretization. Only two cells at both sides of the edge have to be chosen out of the two or more cells that are available at either side of the edge. With the chosen cells, the existing discretization method can be used. Natural questions arise: which cells must be chosen and must we choose the same cells for the advective flux and the viscous flux or not. For the first option, new discretization schemes have to be designed, possibly generalizations or combinations of the existing schemes. Notice that for both options the 6-point edge molecule becomes a \((N_1 + N_2)\)-point edge molecule, though in the second option some matrix coefficients may be zero. As a first step we have chosen to use option 2. We intend to design and implement the first option at a later time.

We are left with the choice of selecting two secondary cells out of the \(N_1 - 1\) cells that are available, where \(N_1\) is the number of edges of cell 1. Also we have to select two cells out of the \(N_2 - 1\) cells that are available in case of cell 2. The two selected cells that are neighbor of cell 1 are denoted by \(A\) and \(B\) where \(A\) is the cell with the lowest cell number of the two selected cells and \(B\) is the other selected cell. Similarly, the two selected cells that are neighbor of cell 2 are denoted by \(C\) and \(D\) where \(C\) is the selected cell with the lowest cell number of the two selected cells. There is no reason why the chosen cells should be the same for the advective and the viscous flux, in fact it appears that the criteria for a suitable choice are different for the two types of fluxes. This concerns both restrictions on the choices and accuracy considerations. We will treat the two types of fluxes independently and take the constraints on the position of the cell centers from Section 3 into account. Finally we will discuss the particularities of the cells at the boundary of the domain. The case of a grid with entirely quadrilaterals will be given special attention, since this type of grid will be used in the numerical experiments.

**Advective flux**

1. We want to keep the positivity property of the advective flux scheme. This means that the cells have to be chosen such that \(n_{1A} \cdot t_{1e} \geq 0\) and \(n_{B1} \cdot t_{1e} \geq 0\) hold, i.e., one of the two selected cells must be above the line through \(t_{1e}\), whereas the other selected cell must be below that line. On average this requirement reduces the available secondary cells with a factor two. It will presumably speed up the selection process accordingly.

2. Suppose \(\mathbf{u} \cdot \mathbf{n} > 0\). For reconstruction of the edge value of the concentration, the gradient of the concentration in \(P_1\) is needed. This gradient is computed on the triangle that is formed by \(P_1\) and the cell centers of the two chosen cells (\(P_A\) and \(P_B\)). It is well known that for accuracy reasons such a triangle should not be too stretched. In particular the component of the gradient in the direction of the narrow side of the stretched triangle may be very inaccurate, see Figure 4. Now, unstructured grids can have a very regular structure in certain regions. A wrong choice of the two selected cells may therefore lead to inaccuracies as described above. Consider the situation of a quadrilateral grid as shown in
Figure 4: The $x$-component of the gradient computed on this triangle may be inaccurate. The stretchedness of the triangle has no negative consequences on the accuracy of the $y$-component of this gradient.

Figure 5. If the cells 3 and 5 are selected as secondary cells, then the gradient in $P_1$ is approximated by the gradient that is defined by the concentration values at the points $P_1$, $P_3$ and $P_5$. The component of this gradient in the direction $t_{1e}$ (the direction that we’re interested in) may be very inaccurate. A better choice for selection in this situation is to choose the cells 3 and 4 or alternatively 4 and 5. In this quadrilateral case the constraints on the positiveness of the discretization scheme as described above, determine which of these two better options must be chosen. The case $u \cdot n < 0$ is similar. The same considerations now hold for cell 2 and the secondary cell belonging to it. We conclude that in case of quadrilateral cells, the selection of the cells is fully determined by accuracy demands. In case of cells with more than four edges, some cells must be selected that satisfy the accuracy demands as described above, and two cells may be selected randomly from these.

**Viscous flux** For the viscous flux the gradient of the concentration at edge $e$ is needed. Recall from Section 3 that this gradient is determined by the area weighted averages of the gradient on the triangles $P_1P_aP_2$ and $P_1P_2P_b$, where the points $P_a$ and $P_b$ are determined by the selection of the secondary cells. As for the advective flux, also here holds that these two triangles should not be too stretched. In case of quadrilateral cells, it seems best to select the cells 3 and 5 as secondary cells belonging to cell 1 and cells 6 and 8 as secondary cells belonging to cell 2, see Figure 5.
Figure 5: A part of the grid. If the cells 3 and 5 are selected, then the approximation of \( \nabla c \) becomes very inaccurate.

5 Numerical experiments

The domain under consideration is known as the Kuststrookmodel, which is a strip of about \( 60 \times 400 \) km in the North Sea along the Dutch coast, see Figure 6. The river Rhine which debouches into the North Sea at \((x, y) = (67, 443)\) km, is an important source of pollution. Numerical experiments by De Kok [1] show that pollutants released at the Rhine outlet drift, forced by the north-east directed residual current, in the direction of the Waddenzee, a protected area in the north of the Netherlands. In our transport computations, field data (water depth, flow velocity) are obtained by waqua, a 2D numerical model for the computation of flow in open waters by the National Institute for Coastal and Marine Management/RIKZ [6]. These flow computations are done on the quadrilateral grid of Figure 6. The forcing that is caused by the salinity gradients is neglected in the WAQUA
computations. The field data are computed with a time step of 120 s and made available after 31 days. These data are taken for one tidal period ([31 days, 31 days + 12.5 hours]) with intervals of 30 minutes, i.e., 25 points of time. For our transport computations on a quadrilateral grid, the same grid is used (20175 cells). Our triangular grid is obtained by a straightforward subdivision of the quadrilaterals (40350 cells). The average edge length is about 1100 m. We use a time step of 1800 s. The average Courant number (average over all grid cells) varies from 2.4 to 3.4 and the maximum Courant number varies from 25 to 93, see Figure 7. The Courant number attains its maximum value almost always near the Dollard in the north-eastern part of the domain.

![Figure 7: The average (left axis) and maximum (right axis) Courant numbers during one tidal period.](image)

It is a non-trivial question how continuous releases must be modeled on a computational grid. Releases on the boundary of the domain are difficult to model because the grid along the boundary is too course to represent local advective flow phenomena. In fact, in the computations, the only significant contribution is the dispersive term. Releases in the interior are difficult as well. Typical releases are delta-like functions that have support of only one grid cell. Numerical diffusion has to be added to prevent numerical problems that may occur because of the large local concentration gradients. In some early experiments we explored a time-independent velocity field and it was found that a release function with support of 5×5 grid cells was necessary to prevent numerical problems. This is a much larger area than the actual physical area of release. However, in later experiments with the fully time-dependent field, a release function with support of only one cell turned out to be possible. It is expected that for other velocity fields with larger residual velocity a release function with support of more grid cells is necessary. Yet another option is to take a continuous release at the interior of the Rhine, with a release function with support of as many cells as needed to prevent numerical problems. Since the residual current is North-Sea directed, this can be considered (from the viewpoint of the North-Sea without the Rhine) as a boundary value at the position of the Rhine inlet with a non-zero velocity.
at the edge. This is not done in the experiments described in this section. In this report, a continuous release of one grid cell is taken, in the North Sea close to the Rhine outlet.

The input velocity fields do not satisfy $\nabla \cdot \mathbf{u} = 0$. On average (over all cells) $\nabla \cdot \mathbf{u} \approx 10^{-4}$ and the maximum is $10^{-3}$. This causes an unwanted source term in the advection-diffusion equation. Therefore, the velocity field is adjusted so as to make the velocity field divergence free per cell.

Figure 8 presents the computed concentration after 72 tidal periods for both the triangular grid method and the quadrilateral grid method. The triangular and quadrilateral methods give equal time scales for the spreading of the concentration. The triangular method gives a more diffusive answer than the quadrilateral method.

Figure 9 gives the history profile of the concentration at a point in the North Sea at $(x, y) = (69, 457) \text{ km}$ for both the triangular and the quadrilateral method. Here one can clearly see that the concentration reaches the checkpoint at the same time. The concentration for the quadrilateral method is larger than for the triangular method because the solution of the quadrilateral method is less diffusive which gives an higher value of the concentration in the center of the constituent plume (where the checkpoint is located).

We also compared these results with the transport module of WAQUA. In Figure 10 the concentration profile after 72 tidal periods is given. We can see that the propagation speed of the concentration along the Dutch coast is about the same for our transport solver and WAQUA. The maximum concentration value is somewhat higher for WAQUA. Note also the occurrence of negative concentrations at the point of release in the WAQUA computations because of the large concentration gradient at that position.
6 Conclusions

In transport computations, a high degree of flexibility is wanted in choosing the structure of the computational grid. This can for example be dictated by the complex geometry of the physical domain or by the demand to be able to model very local processes such as contaminant releases. To increase this flexibility, we focus on the application of hybrid grids. As a first step we have generalized a triangular solver to quadrilateral grids. The same discretization scheme as for the triangular solver is used with per edge a selection of the available cells.

Which cells must be selected is in part determined by accuracy demands. These demands imply that in general different cells must be selected for the advective and the viscous flux. For the advective flux, the requirement of positivity restricts which cells can be selected. For the viscous flux, the accuracy of the Green-Gauss reconstruction plays an
essential role. In the case of quadrilateral grids, these demands uniquely determine which cells must be selected. In the general case, it may be that some cells must still be selected out of more cells that satisfy all demands. This selection may be done then in a random way.

In numerical experiments we have compared the computation of transport on triangular and quadrilateral grids. The results are very similar. Among others, the time scale of spreading of the contaminant is almost equal. Also a comparison with the transport module of WAQUA is done. These results are also very similar. The WAQUA solution however contains significant negative concentrations at the point of release, unlike the solution of the computation with our solver.

In a next step, we want to extend this method to ‘real’ hybrid grids with different types of cells. We also plan to use a hybrid method that uses all secondary cells around an edge instead of selection of two secondary cells at either side of the edge.

References


A The use of hybrid grids in grid refinement

Grid refinement is an important technique for accurate transport computations. For example, grid refinement is useful in areas with high concentration gradients and at points where it is necessary to include information at sub-grid level such as at release points. The grid at these areas may be too coarse for accurate results. Numerical diffusion has to be added to reduce the spurious oscillations, leading to smeared solutions. A better method is to locally refine the grid, so that the gradient is spread over more grid cells.

Unfortunately, grid refinement is difficult to achieve if angle-conserving properties of the grid are put forward. See Figure 11.

![Figure 11: Two methods for cross-section of a triangle.](image)

In 11(a), the cells $\Delta_1$ and $\Delta_2$ are refined by adding the dashed lines as two new edges, i.e., $\Delta_1$ and $\Delta_2$ are both split into two cells. Consequently, the smallest angle of the corners of the new cells is $\alpha/2$. As stated earlier, reduction of angles negatively influences the accuracy of the computations and is therefore not preferable.
Angle reduction can be avoided by the type of grid refinement as in 11(b). The three new edges (dashed lines) split $\Delta_2$ into four new smaller cells that have the same shape and same angles in the corners of the cell as the original cell. This type of grid refinement introduces junctions that are not allowed in triangular grids, i.e., one edge ends at the interior of an other edge. It is allowed however if we consider these cells as belonging to a hybrid grid. We can now define $\Delta_1$ as a quadrilateral cell with one angle of $180^\circ$ at the new junction.

We have to look carefully at the consequences of this procedure, see Figure 12. Because of the much larger size of cell $\Delta_4$ then cell $\Delta_1$ and $\Delta_4$, the triangle $P_1 P_3 P_4$ is very narrow. It is a wrong conclusion however that this type of grid refinement should therefore not be applied. Narrow triangles for Green-Gauss reconstruction can also result from grids with a high degree of irregularity.

Figure 12: An example of the use of hybrid grids in grid refinement.

An other method for avoiding the inaccurate approximation of concentration gradients is to use other cells than only the primary and secondary cells that belong to edge $e$. In terms of Figure 12, we could decide to use triangle $P_1 P_3 P_4$ for Green-Gauss reconstruction. In general, it may be worthwhile to do a careful analysis of the whole grid, which must be used for the Green-Gauss reconstruction procedure. Since this has to be performed only once prior to the time stepping procedure it has only limited costs in terms of computational operations, whereas the benefit in accuracy may be considerable.

Using other cells than only the primary and secondary cells of edge $e$ may have negative consequences for solving the system of equations (more non-zero elements per row in the matrix) and for the complexity of the data structure and computer code. However, part of the increasing complexity has already been dealt with in the extension to hybrid grids.