# A post-processing to restore numerical consistency for the most classical multiple flow direction algorithms

Julien Coatléven, Benoit Chauveau

**Preprint**

# A post-processing to restore numerical consistency for the most classical multiple flow direction algorithms

Julien Coatléven *†, Benoit Chauveau *

**Abstract.** In a recent paper, a consistency correction for the water flux using multiple flow direction (MFD) algorithms that account for exchanges between a cell and its neighbors was proposed, thanks to a reinterpretation of the MFD as a well chosen discretization of the Gauckler-Manning-Strickler continuous equation. Building on those results, we introduce here a general framework allowing to derive consistent expressions of the water flux for the most commonly used multiple/single flow direction (MFD/SFD) water flow routines, including versions in which water is flowing from a node to its neighbors. This general framework is shown to be sufficiently general to encompass the alternative continuous definition of the unit catchment area of the literature. Numerical examples illustrate the consistency and convergence of the proposed water flux reconstructions.

## Introduction

A wide variety of mathematical models describing the flow of water could be used in landscape evolution models (LEMs), depending on the prominent space and time scales considered. The most complete model is undoubtedly the Navier-Stokes model which allows for very precise but prohibitively costly simulations. The shallow-water approximation is sometimes used to solve rivers system (e.g. [2]) or to simulate glacial dynamics [18]. Despite a reduced computational cost compared to the Navier-Stokes model, this model has not often been explicitly deployed in LEMs. Probably one of the reasons is that computationally efficient water flow routing algorithms have been developed during the last decades. Those algorithms are built assuming that the water flow follows the direction of steepest descent (e.g. [33, 22, 20, 41, 28, 42, 40]), and are able to simulate relatively complex water flow networks despite this inherent simplicity. Multiple flow direction (MFD) and single flow direction (SFD) algorithms are among the most known water-flow routing families implemented in reference LEMs such as in SIBERIA([54, 55, 53]), CAESAR-Lisflood ([5, 14]), FastScape ([9]), eSCAPE ([44]), CIDRE ([10]), EROS ([15]) or BadLand ([45]), or in stratigraphic models such as DionisosFlow ([27]). This list being not exhaustive, the reader is referred to [49, 51, 50, 1, 32] for a complete review. Following the terminology of [1], we consider two different families of water flow algorithms, based on their representation of the discretized domain. These are referred to as cell-to-cell and node-to-node algorithms, respectively. The first category includes all models that only consider interactions between a cell and its neighboring cells, while the second includes those that focus on interactions between a node and its neighboring nodes. The main differences between the various algoritms inside each family lie in their representation of the discretized domain (cell-to-cell or node-to-node) and by the empirical choice made to distribute water among the mesh elements.

The empirical foundations of the MFD/SFD water flow routing and their lack of mathematical framework make them very difficult to validate. A first behavior known since a long time is not very encouraging: the water flow distribution $\mathcal{Q}_w$ is mesh dependent. This is probably the most documented problem of the LEM community since more than twenty years (e.g. [46, 35, 1]) and one that still disturbs current models. Smart solutions have been published to minimize this effect without making it completely vanish ([38, 35]). An alternative definition of the specific catchment area often used as a proxy for water flow was proposed in [24, 8], through the discrete

*IFP Énergies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France
†julien.coatleven@ifpen.fr

solution of an abstract uniform flow equation, which effectively solves the anomalous mesh dependency issue. Independently and following another path, in [11] the simplest cell-to-cell MFD algorithms family has been proved to coincide on Cartesian meshes with a classical discretization of the water mass conservation Gauckler-Manning-Strickler model (GMS). The output of the MFD algorithms is shown to correspond exactly to a mesh-dependent mean of the water flux associated with the discrete GMS model. This result not only explains the mesh and numerical dependency since the output of the MFD does not fulfill the consistency criteria, but it also provides a way to correct it in a post-processing step leading to a consistent discrete approximation of the GMS water flux. This new discrete water flow is then as mesh independent as possible, in the usual numerical analysis sense of convergence when the mesh size goes to zero.

The main purpose of this paper is to extend this result to a wider range of classical MFD/SFD algorithms, including the node-to-node versions. Generalizing the results of [11], they can in fact all be corrected in a post-processing step, finally solving the grid dependency issue while keeping the diversity of approaches considered in the literature. To do so, we will need to consider a more general GMS model than that of [11]. This generalized GMS model being also a generalization of the model proposed in [24, 8], this finally closes the loop between MFD algorithms and the specific catchment area defined in [24, 8]. We believe that the resulting easy to implement post-processing step could benefit to numerous existing water flow models built from MFD/SFD algorithms.

The paper will thus be organized as follows: in a first section we recall the continuous specific catchment area model of [24, 8] and introduce our general GMS model, emphasizing their link. Next, we generalize the results of [11] detailing the link between the general GMS model and a wider range of cell-to-cell MFD algorithms, and explain how this allows their correction in a post-processing step. The recovered consistency is then illustrated through numerical examples. In a third section, we extend those results to node-to-node MFD algorithms, and again illustrate on numerical examples how the simple post-processing step we propose allows to recover consistency. We conclude by some remarks on the applicability domain of the GMS model.

## 1 Gauckler-Manning-Strickler models and multiple flow direction algorithms

In principle, the output of a MFD algorithm is the historically loosely defined "local discharge of water" $\mathcal{Q}_w$. The practical computation of $\mathcal{Q}_w$, when not carefully conducted, is the weak point of many models causing them to lose any hope of consistency in the mathematical sense of the term. This is one of the main reasons why we observe mesh dependency is some LEMs. We recall in this section how to define a physically based "local discharge of water" that maintains consistency. More details can be found in [11, 24, 8]. This discussion was essentially already conducted in [12], thus no true originality is claimed here. We have chosen to recall this in full details for the reader's convenience, as well as to fix some vocabulary and notions. Notice that in [12], it was furthermore been shown that an inadequate treatment of the coupling between water flow and sediment transport leads to another kind of artificial mesh dependencies than those we consider here.

Classically, $\mathcal{Q}_w$ is computed directly from the so-called drainage or catchment area $CA$ (also referred as the contributing area). For a given outlet of the topography, it corresponds to the area of the projection on $\Omega$ of the part of the topography from which the water flows to the considered outlet ([31, 30, 8]). Despite being a very intuitive notion, it has evaded for a long time a precise mathematical definition and was only obtained through some algorithmic procedure. Classical multiple flow direction (MFD) algorithms precisely aim at providing a practical way to compute a discrete approximation $CA_\epsilon(K)$ of the catchment area $CA$ for a mesh cell $K$ (where $\epsilon$ stands for the mesh precision), and in this way a discrete approximation $\mathcal{Q}_K$ of $\mathcal{Q}_w$ for cell $K$. As is well documented ([16, 35, 36, 39]) the classical algorithms provide a discrete catchment area $CA_\epsilon(K)$ that depends on the cell size, geometry and orientation with respect to the flow. Several attempts can be found in the literature to reduce this mesh dependency, defining the discrete water flow discharge $\mathcal{Q}_K$ associated to a mesh cell $K$ as $\mathcal{Q}_K = (CA_\epsilon(K)/w(K))$, where $w(K)$ is a normalization factor related to a geometric property of the cell (cf [16]) or to an estimate of the flow width ([35]) defining the so-called (discrete approximation of

the) specific or unit catchment area (SCA/UCA).

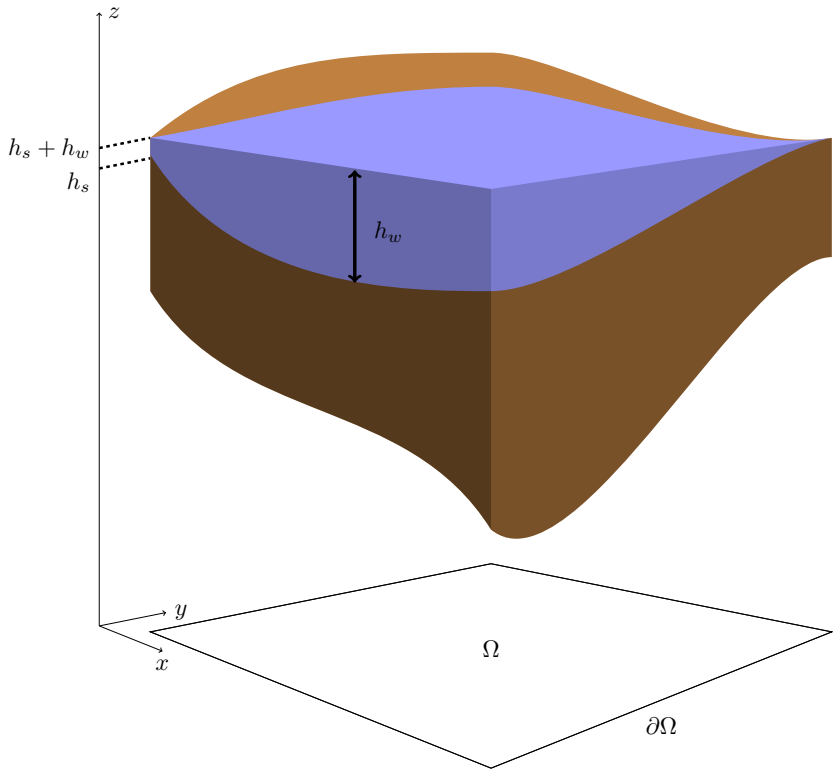



Figure 1: Notations for the topography $h_s$, water height $h_w$ and domain $\Omega$

Recently, a new definition of the specific catchment area $a$ at the continuous level was proposed in [24, 8] in a more modern mathematical way. It consists in solving an abstract uniform flow equation:

$$\left| \begin{array}{ll} -div\left(a\dfrac{\nabla h_s}{||\nabla h_s||}\right) = 1 & \text{in } \Omega, \\ a\dfrac{\nabla h_s}{||\nabla h_s||}\cdot \boldsymbol{n} = 0 & \text{on } \partial\Omega_{in}, \end{array} \right. \tag{1}$$

where the considered fixed geographical region (see figure 1) is modeled by means of a bounded connected domain $\Omega \in \mathbb{R}^2$, $h_s : \Omega \longrightarrow \mathbb{R}$ is the function describing the topography , $\partial\Omega_{in} = \{\boldsymbol{x} \in \partial\Omega \mid \nabla h_s \cdot \boldsymbol{n} > 0\}$ is the part of the boundary that is in going and $\boldsymbol{n}$ denotes the outward normal to $\Omega$. Setting $\mathcal{Q}_w = a$ at the continuous level, this finally leads to compute a consistent discrete approximation $a_K$ of $a$ for a mesh cell $K$. The mesh dependency of $a_K$ is thus reduced to the usual consistency errors of numerical schemes.

Solving (1) instead of resorting to one of the classical MFD algorithms could seem a very different approach at first sight. Indeed, considering for instance the classical cell-to-cell algorithms of [22, 23, 28], one can see that the main principle underlying those MFD algorithms is simply to distribute a fictitious water flow of a mesh cell to the neighboring cells with lower elevation proportionally to a function of the slope, as illustrated in Fig. 2. The main result of [11] is that such a distributing scheme is in fact unexpectedly closely related to a well-chosen

discretization of a water flow model. A full proof is given for the most classical cell-to-cell MFD algorithms (for instance those of [22, 23, 28]) for the following water flow model:

$$\left| \begin{array}{ll} -div\left(k_m h_w^m \nabla h_s\right) = S_w & \text{in } \Omega, \\ k_m h_w^m \nabla h_s \cdot n = B_w & \text{on } \partial\Omega_{in}, \end{array} \right. \tag{2}$$

where $h_w$ is the water height, $m$ a model parameter, $S_w$ is the domain source, and $B_w$ the boundary influx. The coefficient $k_m$ can be thought of as the inverse of a roughness coefficient. It is established in [11] that the discrete quantity distributed by MFD algorithms is a non-consistent mean of the discrete water flux associated with a finite volume approximation of (2), as we detail in next section.

To completely close the gap between MFD algorithms and (1) as well as to encompass a wider range of MFD models of the literature, in the present paper we consider the following generalization of (2), which amounts to a stationary water mass conservation equation with Gauckler-Manning-Strickler (GMS) flux modeling:

$$\left| \begin{array}{ll} -div\left(k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s\right) = S_w & \text{in } \Omega, \\ k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot n = B_w & \text{on } \partial\Omega_{in}, \end{array} \right. \tag{3}$$

where $h_w$ is the water height, $s_{ref} = 1$ m.km$^{-1}$ the reference slope, $p_w$ a model parameter and $\eta_w$ the water mobility function. For simplicity we assume here that the mobility function has no dimension and is a function of $h_w$ only, and that the domain source $S_w$ is given in m$^3$.s$^{-1}$km$^{-2}$ such that its integral over a 2d area measured in km$^2$ coincides with a discharge in m$^3$.s$^{-1}$. The boundary influx $B_w$ is measured in $m^3.s^{-1}km^{-1}$. The coefficient $k_m$ can be thought of as the Strickler coefficient or the inverse of the Gauckler-Manning coefficient up to a change of unit (strictly speaking, this identification is truly valid for channels and if the mobility function $\eta_w$ is equal to a dimensionless hydraulic radius). For this choice of unit for $S_w$, $k_m$ has the unit m.s$^{-1}$ of a speed. Comparing (3) with (1), we see that (1) corresponds to the particular case where $k_m = 1$, $p_w = -1$ and $a = h_w \eta_w(h_w)$, while comparing (3) and (2) we recover (2) by choosing $\eta_w(h_w) = h_w^{m-1}$ and $p_w = 0$. In this sense the GMS model (3) is a generalization of (1) and (2) that allows to include the classical ingredients (non linear slope dependency and some spatial heterogeneity) of the MFD literature. Closely following [11], we explain in the remaining of the paper how model (3) can be related to most of the MFD algorithms of the literature and how it allows to correct them through a simple post-processing step.

**Remark 1.1.** To say that this model uses Strickler coefficients or the inverse of Gauckler-Manning coefficients does not necessarily mean that its scope of application is limited to channels: it depends to the specific choice made on the model parameter values. Steady state analysis ([26, 7]) for channels suggests to use values $\eta_w(h_w) = (h_w/h_{ref})^{1/2}$ and $p_w = -1/2$, while the classical Gauckler-Manning-Strickler formula would coincide with $\eta_w(h_w) = (R_h(h_w)/h_{ref})^{2/3}$ with $R_h(h_w)$ the hydraulic radius and again $p_w = -1/2$. When applied to large time and space scales landscape evolution models, these calibrations are no more valid and at this stage we suggest considering $\eta_w$ and $p_w$ as modeling parameters that can be tuned for each considered problem.

## 2 Obtaining consistent cell-to-cell MFD algorithms through a discretization of the GMS model

As mentioned above, the results of this section are mostly a quite straightforward generalization of the results of [11] on (2) to the more general GMS model (3). We believe that the node-to-node version will be easier to understand after detailing the simpler cell-to-cell one, which is the main motivation for this section.

### 2.1 Mesh description and the classical cell-to-cell MFD algorithms

To make precise statements and establish the correspondence with MFD algorithms, we need to introduce quite a few notations for describing our meshes. Assume that $\Omega$ is a polyhedral bounded connected domain

of $\mathbb{R}^2$, whose boundary is denoted $\partial\Omega = \overline{\Omega}\backslash\Omega$. We recall the usual finite volume notations describing a mesh $\mathcal{M} = (\mathcal{T}, \mathcal{F})$ of $\Omega$. The set of the cells of the mesh $\mathcal{T}$ is a finite family of connected open disjoint polygonal subsets of $\Omega$, such that $\overline{\Omega} = \cup_{K\in\mathcal{T}}\overline{K}$. For any $K \in \mathcal{T}$, we denote by $|K|$ the measure of $K$, by $\partial K = \overline{K}\backslash K$ the boundary of $K$, by $\rho_K$ its diameter and by $\boldsymbol{x}_K$ its barycenter. The set of faces of the mesh $\mathcal{F}$ is a finite family of disjoint subsets of $\mathbb{R}^2$ included in $\overline{\Omega}$ such that, for all $\sigma \in \mathcal{F}$, its measure is denoted $|\sigma|$, its diameter $h_\sigma$ and its barycenter $\boldsymbol{x}_\sigma$ . For any $K \in \mathcal{T}$, the faces of cells $K$ corresponds to the subset $\mathcal{F}_K$ of $\mathcal{F}$ such that $\partial K = \cup_{\sigma\in\mathcal{F}_K}\overline{\sigma}$. Then, for any face $\sigma \in \mathcal{F}$, we denote by $\mathcal{T}_\sigma = \{K \in \mathcal{T} \mid \sigma \in \mathcal{F}_K\}$ the cells of which $\sigma$ is a face. Next, for all cell $K \in \mathcal{T}$ and all face $\sigma \in \mathcal{F}_K$ of cell $K$, we denote by $\boldsymbol{n}_{K,\sigma}$ the unit normal vector to $\sigma$ outward to $K$, and $d_{K,\sigma} = |\boldsymbol{x}_\sigma - \boldsymbol{x}_K|$. The set of boundary faces is denoted $\mathcal{F}_{ext}$, while interior faces are denoted $\mathcal{F}_{int}$. Finally for any $\sigma \in \mathcal{F}_{int}$, whenever the context is clear we will denote by $K$ and $L$ the two cells forming $\mathcal{T}_\sigma = \{K, L\}$, as well as $d_{KL} = |\boldsymbol{x}_K - \boldsymbol{x}_L|$. This for instance allows when looping over the faces $\sigma$ of cell $K$ to denote by $L$ the other face of $\sigma$ without resorting to a too heavy notation. To avoid any confusion with the water height and the topography, $\epsilon = \max_{K\in\mathcal{T}}\rho_K$ will denote the mesh size. For any continuous quantity $u$, its discrete counterpart will be denoted $u_\mathcal{T} = ((u_K)_{K\in\mathcal{T}}, (u_\sigma)_{\sigma\in\mathcal{F}_{ext}})$ where for any $K \in \mathcal{T}$ $u_K$ is the constant approximation of $u$ in cell $K$ while for any $\sigma \in \mathcal{F}_{ext}$ $u_\sigma$ is the constant approximation of $u$ over face $\sigma$.

In the following we will assume that the mesh is orthogonal, i.e. there exists a family of centroids $(\overline{\boldsymbol{x}}_K)_{K\in\mathcal{T}}$ such that:

$$\overline{\boldsymbol{x}}_K \in \mathring{K} \quad \forall K \in \mathcal{T} \quad \text{and} \quad \frac{\overline{\boldsymbol{x}}_L - \overline{\boldsymbol{x}}_K}{|\overline{\boldsymbol{x}}_L - \overline{\boldsymbol{x}}_K|} = \boldsymbol{n}_{K,\sigma} \quad \text{for} \quad \sigma \in \mathcal{F}_{int},\ \sigma = \{K, L\} \tag{4}$$

and let us denote $\overline{\boldsymbol{x}}_\sigma$ the orthogonal projection of $\overline{\boldsymbol{x}}_K$ to the hyperplane containing $\sigma$ for any $\sigma \in \mathcal{F}_K$ and any $K \in \mathcal{T}$ with $\overline{d}_{K,\sigma} = |\overline{\boldsymbol{x}}_K - \overline{\boldsymbol{x}}_\sigma|$, as well as $\overline{d}_{KL} = |\overline{\boldsymbol{x}}_K - \overline{\boldsymbol{x}}_L|$. Then, one can use a two-point finite volume scheme to discretize diffusion operators with scalar diffusion coefficients (no tensors).

In the literature, multiple flow direction algorithms are often considered as purely algorithmic ways of dis-

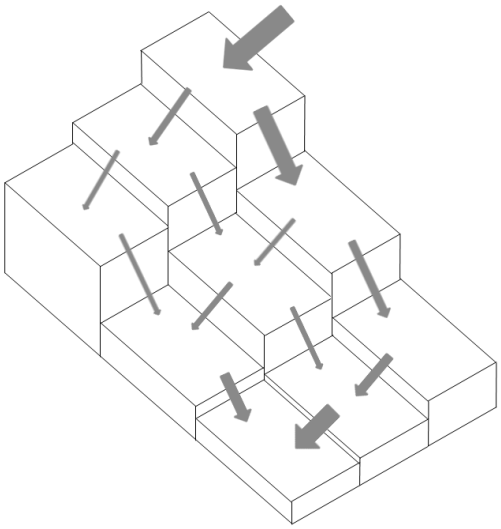


Figure 2: Basic principle of the simplest cell-to-cell MFD algorithm: water is distributed to lower neighboring cells proportionally to the slope (reproduced from [11])

tributing water from one region to another. Thus, they are generally described in a purely algorithmic fashion, although they admit a reformulation as a linear system (first mentioned by [43] although without exhibiting an

explicit formula). Here, we have chosen to introduce the most classical MFD algorithm illustrated on figure (2) following the algebraic formulation of [11] where it is detailed how to obtain it from its algorithmic description. Denoting $h_{s,K} = h_s(\boldsymbol{x}_K)$ for any $K \in \mathcal{T}$ and $h_{s,\sigma} = h_s(\boldsymbol{x}_\sigma)$ for any $\sigma \in \mathcal{F}_{ext}$, from [11] we know that this cell-to-cell MFD algorithm is equivalent to solving the following linear system for the unknown $(\widetilde{q}_K)_{K\in\mathcal{T}}$

$$\widetilde{q}_K = |K|S_{w,K} + \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}<h_{s,L}} \frac{|\sigma|\widetilde{q}_L}{d_{KL}s_L}\left(h_{s,L} - h_{s,K}\right) \quad \forall K \in \mathcal{T} \tag{5}$$

using an ordering for the cells of $\mathcal{T}$ based on decreasing topography $h_{s,K}$ and a lower triangular solver, where $S_{w,K}$ is the source term in cell $K$, the discrete slope $s_{KL}$ is given by

$$s_{KL} = \frac{|\sigma|}{d_{KL}}(h_{s,K} - h_{s,L}) = -s_{LK}$$

and the total positive slope $s_K$ of cell $K$ is given by

$$s_K = \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}\geqslant h_{s,L}} \frac{|\sigma|}{d_{KL}}\left(h_{s,K} - h_{s,L}\right)$$

Indeed, the principle of the most classical distribution formula is simply to distribute the "flow" $\widetilde{q}_L$ in cell $L$ to the lower, neighouring cells $K$ proportionally to the ratio $s_{KL}/s_L$ of the discrete slope $s_{LK}$ between the high cell $L$ and the low cell $K$ regarding the total positive slope $s_L$ of the high cell $L$. This is precisely the meaning of formula (5) but in a reverse fashion, since it says that the total "flow" $\widetilde{q}_K$ in cell $K$ is equal to the local source terms and the flow coming from the higher neighboring cells $L$. Notice that in many cases of the literature, since the MFD algorithm is applied on a uniform Cartesian mesh with the same space step in each direction, the face measure $|\sigma|$ is simply omitted (see for instance [22, 23, 28]), with no effect on the ratio $s_{KL}/s_K$. It is not difficult to generalize the algebraic formulation (5) to more advanced MFD algorithms, for instance those using a power of the slope. Here we recover such generalizations through the equivalence with a discretization of the general GMS model. When the water source $S_{w,K}$ is chosen constant equal to one, the unknown $\widetilde{q}_K$ of the MFD algorithms is used to defined a discrete catchment area $CA_\epsilon(K)$ for cell $K$ by setting $CA_\epsilon(K) = \widetilde{q}_K$. As explained in the previous section, the discrete "local discharge of water" $\mathcal{Q}_K$ associated is computed by normalizing as $\widetilde{q}_K$ i.e. setting $\mathcal{Q}_K = \widetilde{q}_K/w(K)$, where $w(K)$ is a normalization factor. Two very common normalizations are the diameter of the cell $w(K) = \rho_K$ ([16]) or the effective flow length in the cell, which is defined as the length of segment defined by the intersection between the cell and the line going through the center of the cell and oriented following the slope $\nabla h_s$ ([35]).

## 2.2 Cell-to-cell MFD algorithms and the GMS model

The starting point of a finite volume discretization is to integrate equation (3) over each cell $K$:

$$-\int_K div\left(k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w}\nabla h_s\right) = \int_K S_w.$$

Denoting $S_{w,K} = S_w(\overline{\boldsymbol{x}}_K)$ and using Stokes' formula, this leads to:

$$-\sum_{\sigma\in\mathcal{F}_K}\int_\sigma k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w}\nabla h_s \cdot \boldsymbol{n}_{K,\sigma} = |K|S_{w,K}.$$

To handle boundary terms, if one further assumes that $h_{s,\sigma} \geqslant h_{s,K}$ for any $\sigma \in \mathcal{F}_{ext}$ and $K \in \mathcal{T}_\sigma$ which amounts to assuming that water cannot leave the computational domain and is generally what is done in usual

presentation of MFD algorithms, then for $\sigma \in \mathcal{F}_K \mathcal{F}_{ext}$ we get:

$$-\int_\sigma k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n} \approx -|\sigma| B_{w,\sigma},$$

with $B_{w,\sigma} = \frac{1}{|\sigma|}\int_\sigma B_w$ if $h_{s,\sigma} > h_{s,K}$ and 0 otherwise. Choosing a finite volume scheme then simply amounts to choosing how to approximate each term appearing in the face integrals. The most natural and classical finite volume scheme consists in choosing constant approximate values $k_{m,\sigma}$ and $\boldsymbol{G}_{s,\sigma}$ for $k_m$ and $||\nabla h_s||^{pw}$ along each face $\sigma$ and to use an upwind scheme $h_{w,\sigma}^{up}$ for the true unknown $h_w \eta_w(h_w)$:

$$-\int_\sigma k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n} \approx -k_{m,\sigma} s_{ref}^{-p_w} ||\boldsymbol{G}_{s,\sigma}||^{p_w} h_{w,\sigma}^{up} \int_\sigma \nabla h_s \cdot \boldsymbol{n}.$$

Finally, thanks to our hypothesis on mesh orthogonality we can use the two-point flux approximation (TPFA) to compute $\int_\sigma \nabla h_s \cdot \boldsymbol{n}$. The TPFA consists in noticing that for a linear function $h_s$, the gradient being constant and satisfying $\nabla h_s \cdot \boldsymbol{n}_{K,\sigma} = \frac{1}{\overline{d}_{KL}}(h_s(\boldsymbol{x}_L) - h_s(\boldsymbol{x}_K))$, the following formula:

$$-\int_\sigma \nabla h_s \cdot \boldsymbol{n} = -\nabla h_s \cdot \int_\sigma \boldsymbol{n} = \frac{|\sigma|}{\overline{d}_{KL}}(h_s(\boldsymbol{x}_K) - h_s(\boldsymbol{x}_L)),$$

is exact since $\frac{1}{\overline{d}_{KL}}(\boldsymbol{x}_L - \boldsymbol{x}_K) = \boldsymbol{n}_{K,\sigma}$ and will thus be a first order approximation of the flux. More precisely, denoting $h_{w,K}$ for any $K \in \mathcal{T}$ the discrete water height value associated to cell $K$, for any $K \in \mathcal{T}$ the proposed finite volume scheme rewrites:

$$\sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}} \tau_{KL} h_{w,\sigma}^{up} (h_{s,K} - h_{s,L}) = |K| S_{w,K} + \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}, h_{s,K} < h_{s,\sigma}} |\sigma| B_{w,\sigma},$$

where the upwind value is given by $h_{up}^{w,\sigma} = h_{w,K}\eta_w(h_{w,K})$ if $h_{s,K} \geqslant h_{s,L}$ and $h_{up}^{w,\sigma} = h_{w,L}\eta_w(h_{w,L})$ if $h_{s,K} < h_{s,L}$, the transmissivity $\tau_{KL}$ is given by:

$$\tau_{KL} = \frac{|\sigma| k_{m,\sigma}}{\overline{d}_{KL} s_{ref}^{-p_w}} ||\boldsymbol{G}_{s,\sigma}||^{p_w},$$

and where $\boldsymbol{G}_{s,\sigma} = \frac{1}{2}(\boldsymbol{G}_{s,K} + \boldsymbol{G}_{s,L})$ and $\boldsymbol{G}_{s,K}$ is a discrete approximation of the gradient of $h_s$ in cell $K$. If the gradient is known, we can simply take:

$$\boldsymbol{G}_{s,K} = \nabla h_s(\boldsymbol{x}_K)$$

If only the pointwise values of $h_s$ are known, we use a discrete reconstruction of the gradient. To derive it, we use:

$$\mathbb{I}_d = \sum_{\sigma \in \mathcal{F}_K} |\sigma| (\boldsymbol{x}_\sigma - \boldsymbol{x}_K) \boldsymbol{n}_{K,\sigma}, \tag{6}$$

with $\mathbb{I}_d$ the identity matrix in dimension $d$, leading to

$$\boldsymbol{G}_{s,K} = \sum_{\sigma \in \mathcal{F}_K} |\sigma| \boldsymbol{G}_{s,K} \cdot \boldsymbol{n}_{K,\sigma} (\boldsymbol{x}_\sigma - \boldsymbol{x}_K),$$

and thus on the orthogonal meshes we consider here as by consistency $|\sigma| \boldsymbol{G}_{s,K} \cdot \boldsymbol{n}_{K,\sigma} \approx \int_\sigma \nabla h_s \cdot \boldsymbol{n}_{K,\sigma}$, $\boldsymbol{G}_{s,K}$ is naturally given by:

$$\boldsymbol{G}_{s,K} = \frac{1}{|K|} \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}} \frac{|\sigma|}{\overline{d}_{KL}} (h_{s,L} - h_{s,K})(\boldsymbol{x}_\sigma - \boldsymbol{x}_K)$$

$$+ \frac{1}{|K|} \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}} \frac{|\sigma|}{\overline{d}_{K\sigma}} (h_{s,\sigma} + b_\sigma - h_{s,K})(\boldsymbol{x}_\sigma - \boldsymbol{x}_K).$$

From the mathematical point of view, a natural choice for the face value $k_{m,\sigma}$ is the harmonic mean:

$$k_{m,\sigma} = \frac{\overline{d}_{KL} k_{m,K} k_{m,L}}{k_{m,K}\overline{d}_{L,\sigma} + k_{m,L}\overline{d}_{K,\sigma}} \quad \text{with for instance} \quad k_{m,K} = \frac{1}{|K|}\int_K k_m \ \ \forall\, K \in \mathcal{T},$$

but many other choices are possible. Let us now recall the elementary proof given in [11]: gathering the faces by upwinding kind, we get:

$$\sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}\geqslant h_{s,L}} \tau_{KL} h_{w,K}\eta_w(h_{w,K})\,(h_{s,K}-h_{s,L}) -$$

$$\sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}<h_{s,L}} \tau_{KL} h_{w,L}\eta_w(h_{w,L})\,(h_{s,L}-h_{s,K}) = |K|S_{w,K} + \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{ext}, h_{s,K}<h_{s,\sigma}} |\sigma| B_{w,\sigma}. \tag{7}$$

Setting

$$s_K = \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}\geqslant h_{s,L}} \tau_{KL}\,(h_{s,K}-h_{s,L}),$$

and noticing that $s_L > 0$ as soon as there exists $\sigma \in \mathcal{F}_L \cap \mathcal{F}_{int}$ such that $h_{s,L} > h_{s,K}$, we see that equation (7) can be rewritten:

$$s_K h_{w,K}\eta_w(h_{w,K}) - \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}<h_{s,L}} \tau_{KL} h_{w,L}\eta_w(h_{w,L})\,(h_{s,L}-h_{s,K}) = |K|S_{w,K}.$$

Defining the water outflux by $\widetilde{q}_K = s_K h_{w,K}\eta_w(h_{w,K})$, we thus obtain:

$$\widetilde{q}_K - \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}<h_{s,L}} \tau_{KL}\frac{\widetilde{q}_L}{s_L}\,(h_{s,L}-h_{s,K}) = |K|S_{w,K} + \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{ext}, h_{s,K}<h_{s,\sigma}} |\sigma| B_{w,\sigma}. \tag{8}$$

Using the definition of $\tau_{KL}$, we clearly recover (5) if we choose $k_m = 1$, $p_w = 0$ and $B_w = 0$, and this shows the equivalence between the MFD algorithms and the GMS model is established.

This equivalence between the classical MFD and the two-point flux approximation of the GMS allows to give a continuous interpretation and generalization $CA(\mathcal{O})$ for any region $\mathcal{O}$ of the discrete $CA_\epsilon(K)$ that is computed by MFD algorithms only for mesh cells $K$. Indeed, from $\widetilde{q}_K = s_K h_{w,K}\eta_w(h_{w,K})$ and the consistency of the two-point formula, we see that $\widetilde{q}_K$ approximates:

$$CA_\epsilon(K) = \widetilde{q}_K \approx \sum_{\sigma\in\mathcal{F}_K}\int_\sigma h_w\eta_w(h_w)\left(-k_m s_{ref}^{-p_w}||\nabla h_s||^{p_w}\nabla h_s\cdot\boldsymbol{n}_{K,\sigma}\right)^+.$$

This naturally leads us to define a continuous catchment area $CA(\mathcal{O})$ for any region $\mathcal{O}$ by setting:

$$CA(\mathcal{O}) = \int_{\partial\mathcal{O}} h_w\eta_w(h_w)\left(-k_m s_{ref}^{-p_w}||\nabla h_s||^{p_w}\nabla h_s\cdot\boldsymbol{n}\right)^+, \tag{9}$$

where $h_w$ is the solution of (3) with $S_w = 1$ and where we have denoted $v^+$ the positive part of $v$ (i.e. $v^+ = max(0, v)$). As a by-product, we see that we can indeed interpret the discrete catchment area $CA_\epsilon(K)$ computed through the classical cell-to-cell MFD algorithms as the total flux leaving cell $K$ of a fictitious water flow with a uniform water source $S_w = 1$. Unfortunately, we also see that even at the continuous level, $CA(\mathcal{O})$ strongly depends on the geometry of $\mathcal{O}$ and its orientation with respect to the flow. As a result when the discrete catchment area $CA_\epsilon(K)$ computed from MFD algorithms computing is used to estimate the discrete "local discharge of water" $\mathcal{Q}_K$, it produces cell and thus mesh dependency in the simulated surface water distribution. To overcome this mesh dependency, since $CA(\mathcal{O})$ is the total flux leaving $\mathcal{O}$, it is natural to

try to define a specific catchment area (SCA) by rescaling the CA. The correct scaling would be to set the normalization factor $w$ to the length of the portion of $\partial\mathcal{O}$ along which the fictitious water flow is leaving $\mathcal{O}$. A corrected definition of the specific catchment at the continuous level in the spirit of [16, 35, 36] area would thus be to use:

$$SCA(\mathcal{O}) = \left(\int_{\partial\mathcal{O}} \chi_{-k_m s_{ref}^{-p_w}||\nabla h_s||^{p_w}\nabla h_s\cdot\boldsymbol{n}>0}\right)^{-1} \int_{\partial\mathcal{O}} h_w\eta_w(h_w)\left(-k_m s_{ref}^{-p_w}||\nabla h_s||^{p_w}\nabla h_s\cdot\boldsymbol{n}\right)^{+}, \tag{10}$$

where $\chi$ is the indicator function (i.e. the function with value 1 when the condition is satisfied and 0 otherwise). Depending on the orientation of the flow, the discrete counterpart

$$SCA_\epsilon(K) = \left(\sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}\geqslant h_{s,L}} |\sigma|\right)^{-1} \widetilde{q}_K,$$

of such a normalization will sometimes match the choices of [16] or [35, 36] explaining their partial success. This continuous $SCA$ scales as an approximation of the continuous water flux magnitude:

$$q_w = |k_m h_w \eta_w(h_w)|\, s_{ref}^{-p_w}||\nabla h_s||^{p_w+1}, \tag{11}$$

(in $\mathrm{m^3 s^{-1} km^{-1}}$) but is not equal to it. The SCA defined by (10) is in fact a mean of $q_w$ along the outflow portion of $\partial\mathcal{O}$, and thus still retains some dependency in the geometry of $\mathcal{O}$ and its orientation with respect to the flow. Thus, to obtain a mesh independent quantity it is much simpler and more natural to consider directly the water flow magnitude $q_w$ rather than the SCA. This is precisely the consistency correction proposed in [11]: to use a discrete version of $q_w$ instead o $CA_\epsilon(K)$ or $SCA_\epsilon(K)$. Another strong argument for doing so is to notice that the specific catchment area $a$ of model (1) can be reinterpreted through (3) as computing $q_w$ since:

$$q_w = |k_m h_w \eta_w(h_w)|\, s_{ref}^{-p_w}||\nabla h_s||^{p_w+1} = |a|\ \ ||\nabla h_s||^{-1+1} = a,$$

as we have set $a = h_w\eta_w(h_w) \geqslant 0$, $p_w = -1$, $k_m = 1$ and $s_{ref} = 1$ to merge (1) inside (3). In this way, the consistency correction of [11] for MFD algorithms is another path to recover the conclusions of (1), since $q_w$ is a generalization of $a$ to more complex water flow models.

The MFD formulation allows in turn some interesting observations for the Gauckler-Manning-Strickler model: it is indeed clear that the choice of the water mobility function $\eta_w$ has no influence on the water flux strength $q_w$, as it appears nowhere in (8) and (12). In the same way, only the contrasts of the coefficient $k_m$ will impact $q_w$, as only ratios $\tau_{KL}/s_K$ are appearing in (8) and (12).

## 2.3 Consistency post-processing for cell-to-cell MFD algorithms

To effectively compute an accurate discrete water flux magnitude $q_K$ for each cell $K \in \mathcal{T}$, from [11] we know that we can reconstruct cellwise the water flux vector using (6) by setting:

$$\boldsymbol{Q}_K = \sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}>h_{s,L}} \frac{\tau_{KL}\widetilde{q}_K}{|K|s_K}(h_{s,K}-h_{s,L})(\boldsymbol{x}_\sigma-\boldsymbol{x}_K) -$$

$$\sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}, h_{s,K}<h_{s,L}} \frac{\tau_{KL}\widetilde{q}_L}{|K|s_L}(h_{s,L}-h_{s,K})(\boldsymbol{x}_\sigma-\boldsymbol{x}_K), \tag{12}$$

and simply deduce a consistent water flux magnitude by setting

$$q_K = ||\boldsymbol{Q}_K||. \tag{13}$$

The convergence of the consistent water flux magnitude $q_K$ was rigorously established and illustrated in [11] for the simpler model (2), up to providing error estimates. A straightforward although rather involved adaptation of the proof in [11] would undoubtedly provide the convergence of $q_K$ to the continuous flux when the mesh size $\varepsilon$ goes to zero. In the present paper we limit ourselves to numerical confirmations of the convergence of $q_K$ in the general setting.

We believe that the use of $\widetilde{q}_K$ or its approximately normalized versions instead of $q_K$ in the literature is the main reason why such a strong mesh dependency was observed, without any significant improvement with mesh refinement. Thus, it is important to use $q_K$ instead of $\widetilde{q}_K$ as is the correct output of a MFD algorithm, i.e. using $\mathcal{Q}_K = q_K$ and not $\widetilde{q}_K$ or $\widetilde{q}_K/w(K)$. Notice that from (12) and (13), it is obvious that computing $q_K$ can certainly be considered as a post-processing consistency correction for $\widetilde{q}_K$, that should be easy to implement in legacy software.

For completeness let us finally mention that the assumption that $h_{s,\sigma} = h_{s,K}$ for any $\sigma \in \mathcal{F}_{ext}$ and $K \in \mathcal{T}_\sigma$ is not mandatory and was done only to exactly match the MFD linear system and simplify the presentation. For open boundaries, since this assumption corresponds to assuming that water cannot leave the computational domain, it suffices to modify the value of $s_K$:

$$s_K = \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}, h_{s,K} \geqslant h_{s,L}} \tau_{KL} \left(h_{s,K} - h_{s,L}\right) + \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}, h_{s,K} \geqslant h_{s,\sigma}} \tau_{K\sigma} \left(h_{s,K} - h_{s,\sigma}\right),$$

as well as the definition of $\boldsymbol{Q}_K$:

$$\begin{aligned}
\boldsymbol{Q}_K = & \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}, h_{s,K} > h_{s,L}} \frac{\tau_{KL}\widetilde{q}_K}{|K| s_K} (h_{s,K} - h_{s,L})(\boldsymbol{x}_\sigma - \boldsymbol{x}_K) \\
& + \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}, h_{s,K} > h_{s,\sigma}} \frac{\tau_{K\sigma}\widetilde{q}_K}{|K| s_K} (h_{s,K} - h_{s,\sigma})(\boldsymbol{x}_\sigma - \boldsymbol{x}_K) \\
& - \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}, h_{s,K} < h_{s,L}} \frac{\tau_{KL}\widetilde{q}_L}{|K| s_L} (h_{s,L} - h_{s,K})(\boldsymbol{x}_\sigma - \boldsymbol{x}_K),
\end{aligned} \tag{14}$$

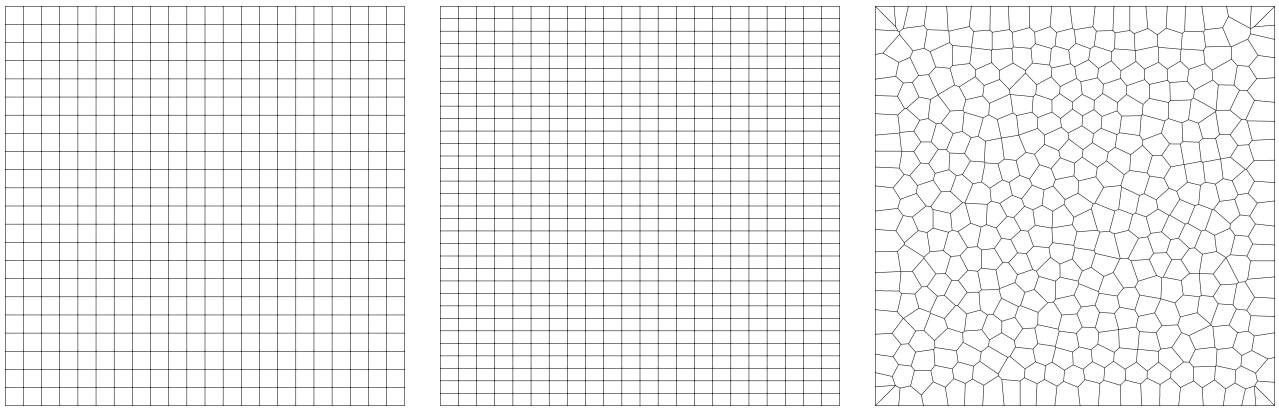
with of course in both cases:

$$\tau_{K\sigma} = \frac{|\sigma| k_{m,\sigma}}{\overline{d}_{K\sigma} s_{ref}^{-p_w}} ||\boldsymbol{G}_{s,\sigma}||^{p_w}.$$

## 2.4 Numerical results for the corrected cell-to-cell MFD algorithms

The consistency post-processing (12)-(13) for MFD algorithms precisely coincides with the replacement of the computation of $\widetilde{q}_K$ or $\widetilde{q}_K/w(K)$ for a mesh cell $K$ by a consistent discrete reconstruction $q_K$ of $q_w$ in each cell $K$. Thus, apart from the usual discretization error no anomalous mesh dependency should remain in $q_K$ in practice, contrary to what is observed for $\widetilde{q}_K/w(K)$ (i.e. $SCA_\epsilon(K)$) given by MFD algorithms. On the contrary, since the quantity $\widetilde{q}_K$ approximates the outflux of a cell it is proportional to the perimeter of a cell, the only convergence that can be expected for $\widetilde{q}_K$ is to zero, while the behavior of $\widetilde{q}_K/w(K)$ will strongly depend on the choice of the normalization $w(K)$. The main purpose of this subsection is to illustrate the behavior of those quantities through some easy to analyze numerical examples. Since it has no impact on the water flux $q_w$ which is the main target of MFD algorithms, for simplicity we consider only the case of constant water mobility $\eta_w(h_w) = 1$. We consider two reference configurations. On the first one we focus on the consistency and convergence of $q_K$ while illustrating that $\widetilde{q}_K$ even rescaled by the attempted values of $w(K)$ from the literature cannot be a good approximation of the flux. Simultaneously, we show that the method can easily handle heterogeneous speed coefficients $k_m$. The second configuration allows us to assess the robustness of the method even when the slope is non-linear ($p_w > 0$). To study the mesh dependency of the results, we consider three sequences of meshes,

all satisfying the orthogonality requirement (4). Namely we use two sequences of Cartesian meshes, the first one with square cells and the second one with rectangular cells, and a sequence of Voronoï meshes. Those three sequences are illustrated on figure 3.



(a) Cartesian mesh: square cells (b) Cartesian mesh: rectangular cells (c) Voronoi mesh

Figure 3: Three sequences of meshes designed to study the mesh dependency within the numerical analyses presented in section 2.4

### 2.4.1 First test configuration: Gaussian topography and Gaussian water height

The domain $\Omega$ is the square $\Omega = ]0, L[\times]0, L[$, the exact topography is given by

$$h_s(x, y) = h_{s,0} \exp\left(-\delta\left((x - x_0)^2 + (y - y_0)^2\right)\right)$$

where $x_0 = y_0 = L/2$ and the exact water height is given by

$$h_w(x, y) = h_{w,0} \exp\left(-\alpha\left((x - x_0)^2 + (y - y_0)^2\right)\right)$$

In practice we will set $L = 1$, $h_{s,0} = 1$ and $h_{w,0} = 1$. For this first configuration, we will consider three different values of the speed coefficient $k_m$, with increasing order of difficulty. The first one is simply the constant case $k_m = 1$, the second one is a sinusoidal case:

$$k_m = 1 + \beta \sin(\omega\pi x) \sin(\omega\pi y)$$

with $\beta = 0.1$ and $\omega = 3$, and the final one is a Perlin noise perturbation :

$$k_m = 1 + \beta\, \theta(x, y)$$

where the function $\theta(x, y)$ is a Perlin noise [37] and the resulting coefficient $k_m$ is depicted on figure 4. Since $q_w$ depends on $k_m$, using these heterogeneous speed coefficients is a simple way to obtain not overly simplistic values for the water flux magnitude $q_w$. Finally, we fix the slope exponent $p_w$ to zero for this configuration.

From [11], we know that $q_K$ should converge towards $q_w$ in $L^2$ norm. We display on figure 5 the corresponding convergence curves, which confirm that we recover the expected convergence. To further emphasize the interest of using $q_K$ instead of $\widetilde{q}_K/w(K)$, on figures 6 to 14 we display for each combination of mesh sequence and value

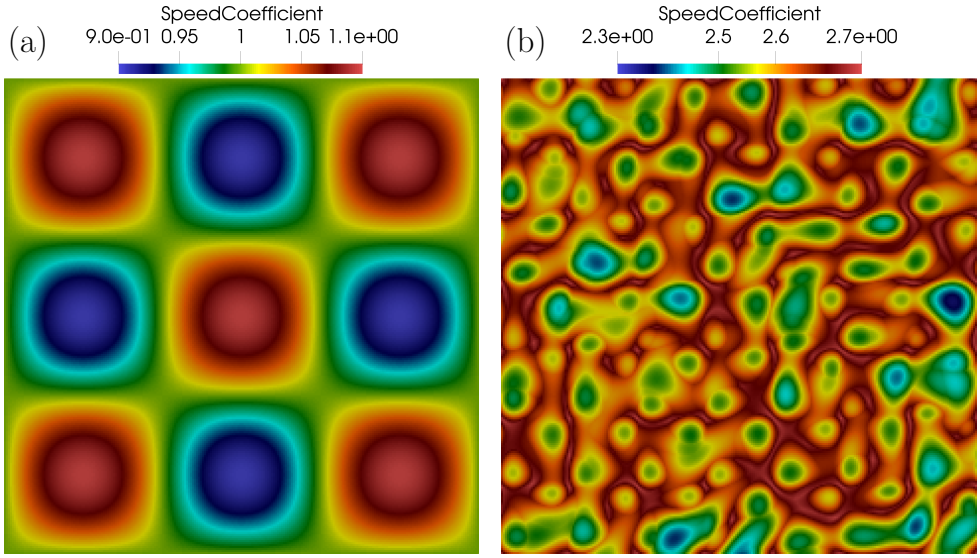



Figure 4: The sinusoidal and perlin noised-based speed coefficients $k_m$

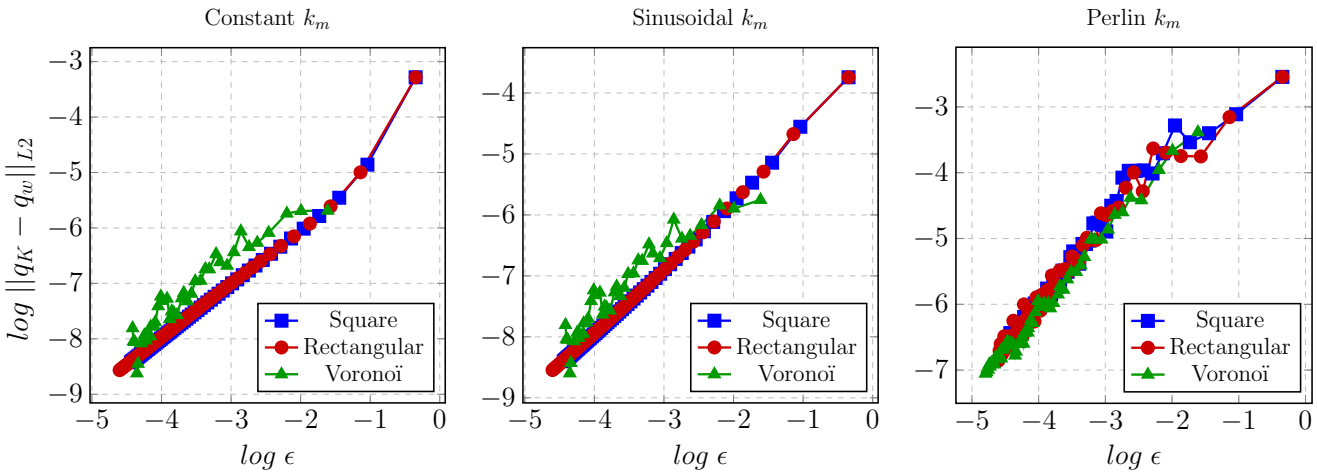



Figure 5: Convergence curves for $q_K$ towards $q_w$ for the first test configuration, for each mesh sequence and each choice for $k_m$

of $k_m$, the exact water flux magnitude $q_w$, the discrete water flux magnitude $q_K$, and $\widetilde{q}_K/w(K)$ for the two choices of normalization $w(K)$. We display those four values for the finest mesh of each sequence.
We clearly see that $q_K$ is a good approximation of $q_w$. However, the mesh dependency of $\widetilde{q}_K/w(K)$ appears on each of those example. The normalization $w(K) = \rho_K$ leads to the correct value range for $\widetilde{q}_K/w(K)$ when compared to $q_K$ or $q_w$. The effective length normalization gives the correct order of magnitude, but not exactly the correct value range. The normalization $w(K) = \rho_K$ leads to a very strong mesh dependency, and a value

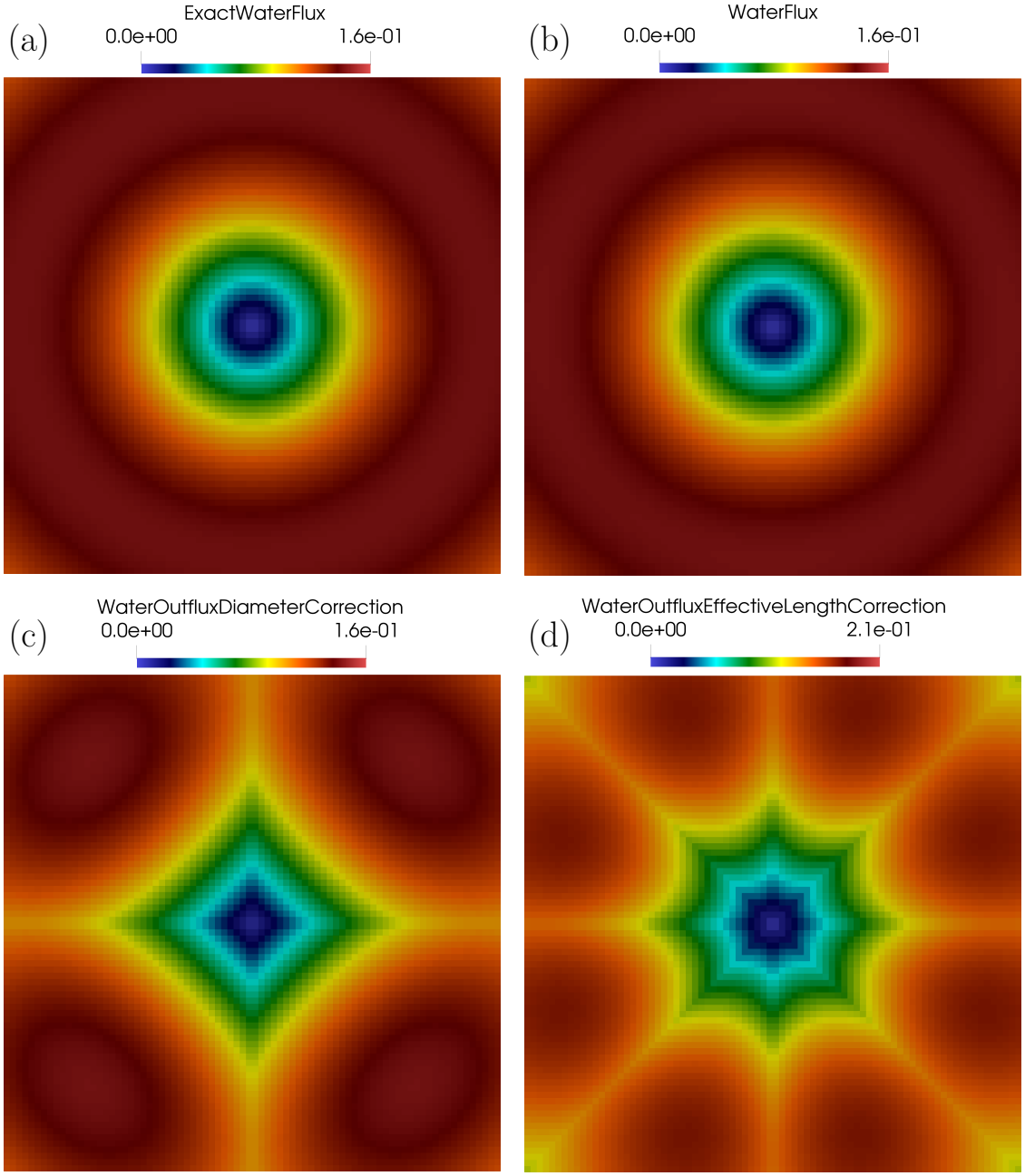



Figure 6: Water flux (top view) obtained for constant $k_m$ on the finest Cartesian mesh with square cells. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

$\widetilde{q}_K/w(K)$ very far from the correct one. On the two Cartesian mesh sequences for each value of $k_m$ (figures

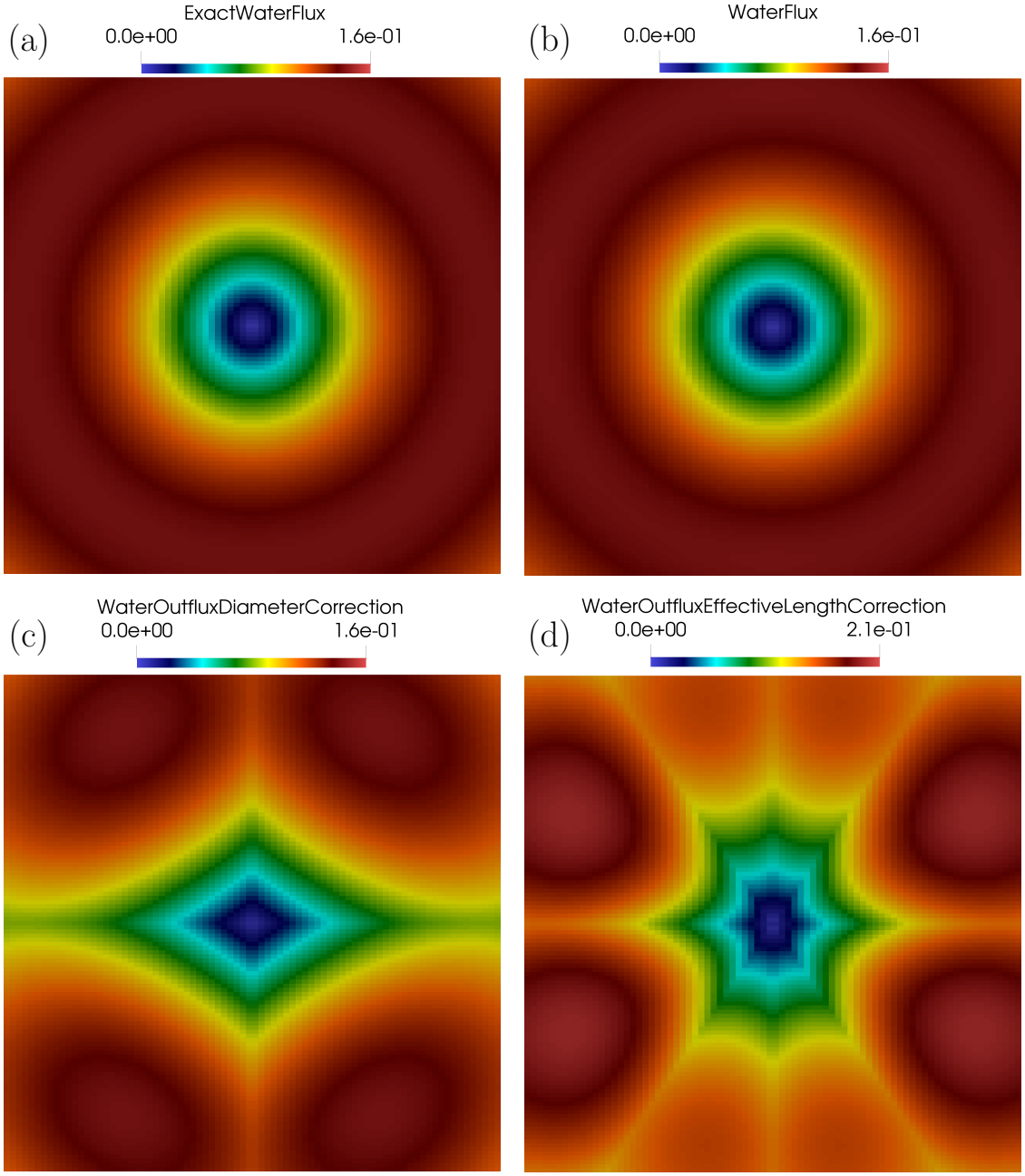



Figure 7: Water flux (top view) obtained for constant $k_m$ on the finest Cartesian mesh with rectangular cells. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

6, 7, 9, 10, 12, 13), we see that $\widetilde{q}_K/w(K)$ is strongly biased in the main directions of the mesh (the two axis

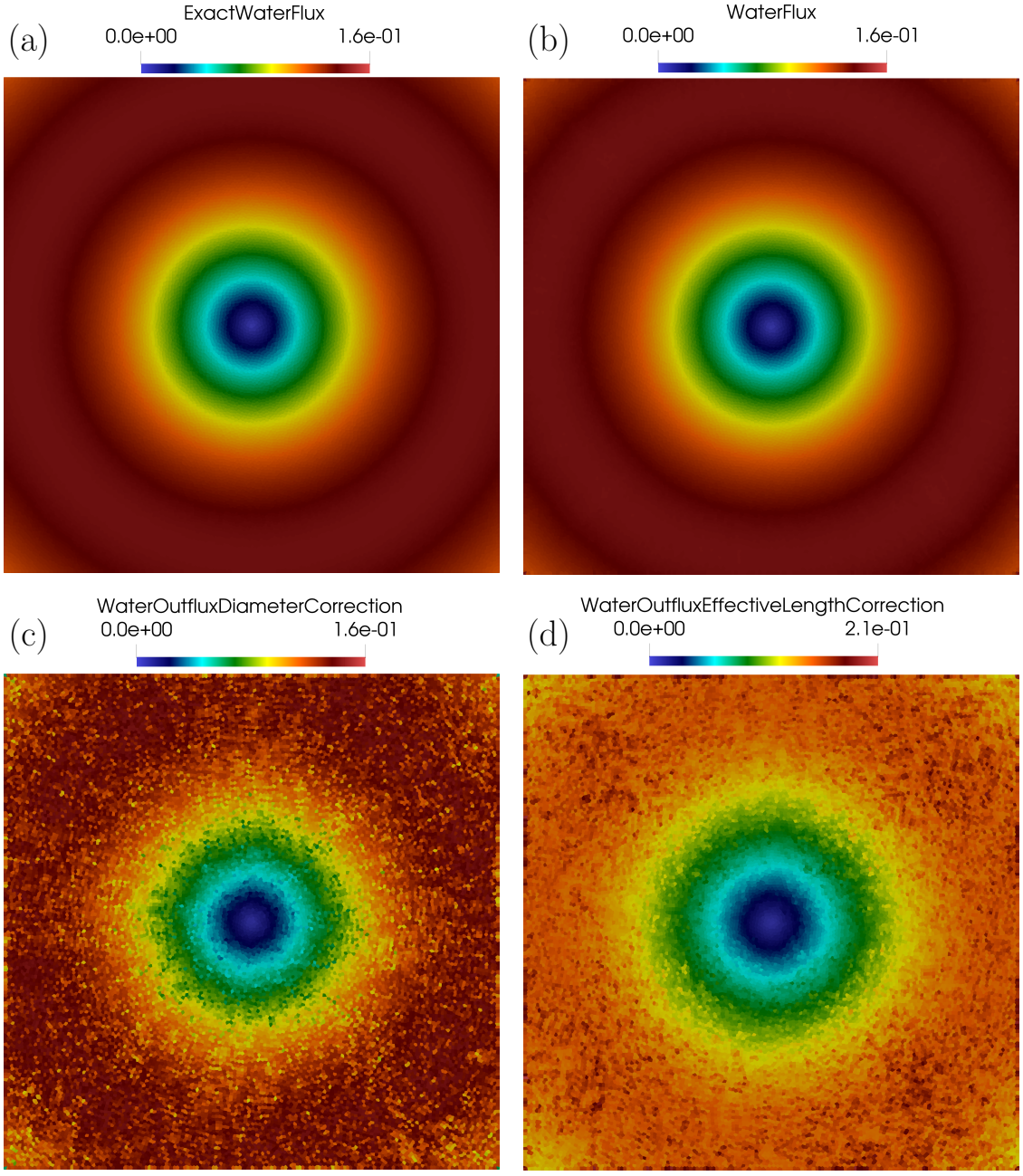



Figure 8: Water flux (top view) obtained for constant $k_m$ on the finest Voronoï mesh. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

$x$ and $y$), and comparing the two sequences we clearly obtain different solutions despite the fact that the two

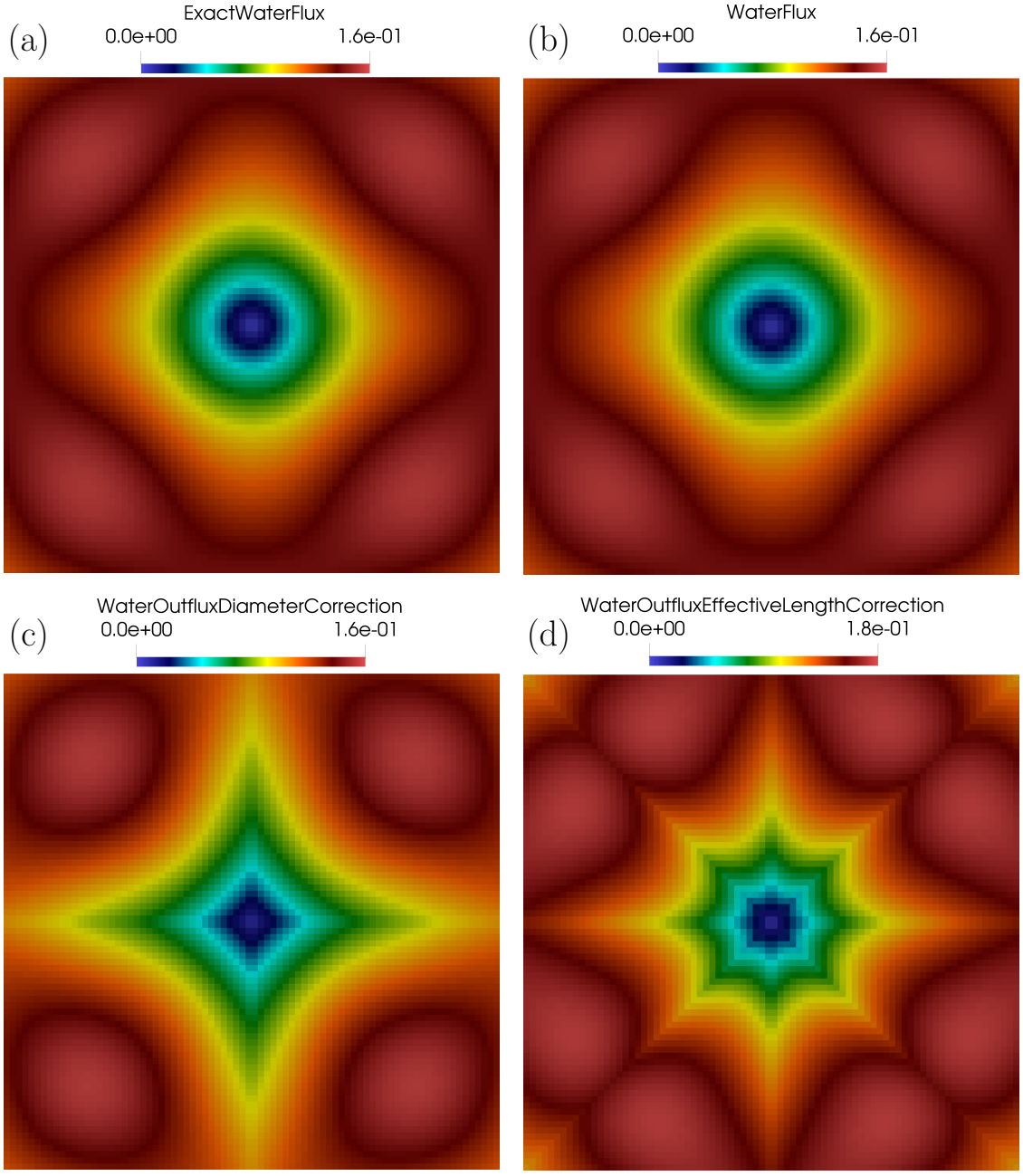



Figure 9: Water flux (top view) obtained for sinusoidal $k_m$ on the finest Cartesian mesh with square cells. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

mesh sequences only slightly differ by the shape of the cells. On the Voronoï mesh sequence (figures 8, 11, 14),

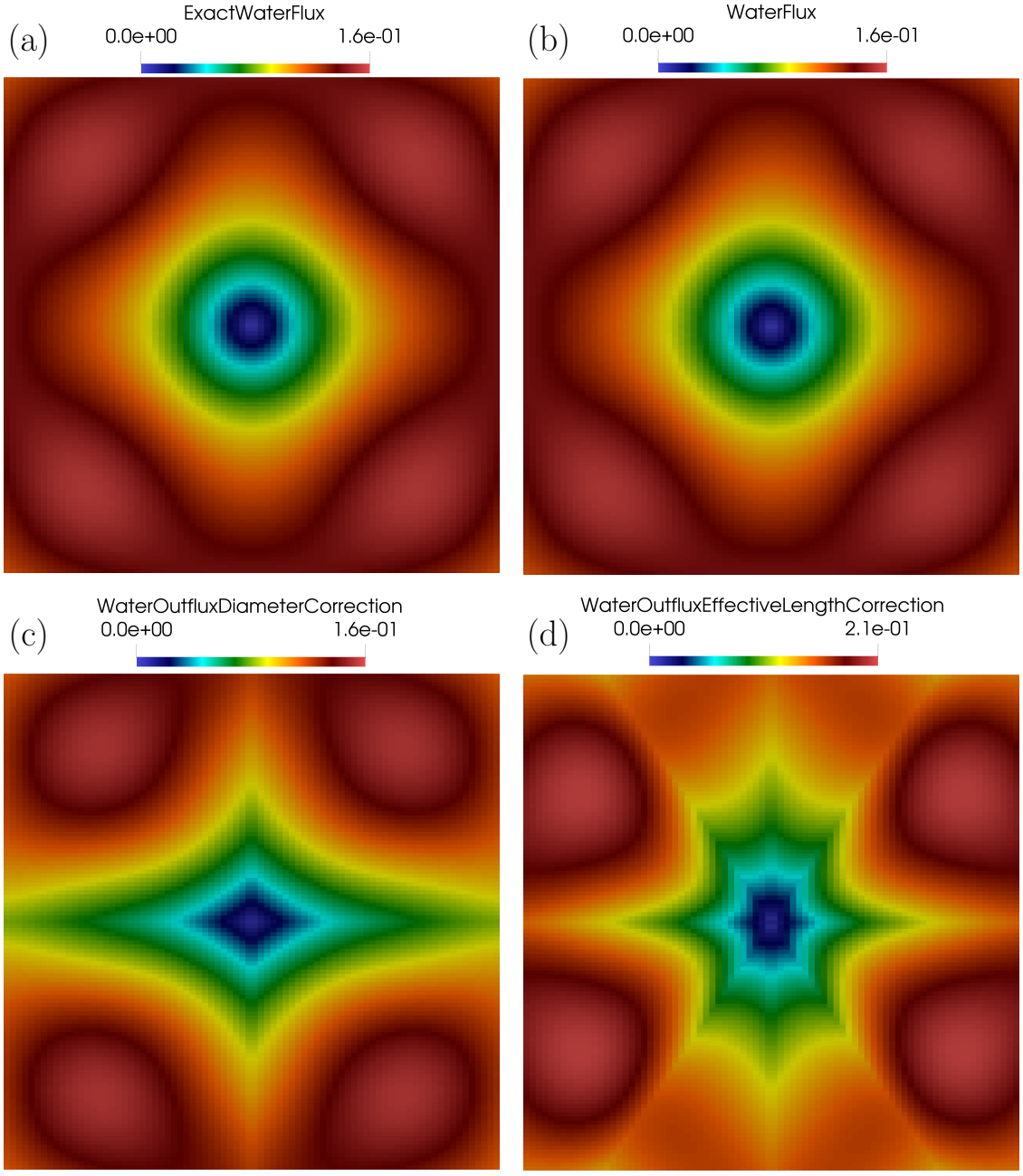



Figure 10: Water flux (top view) obtained for sinusoidal $k_m$ on the finest Cartesian mesh with rectangular cells. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

since the mesh is much less biased towards any specific direction, we start to see the correct solution shape

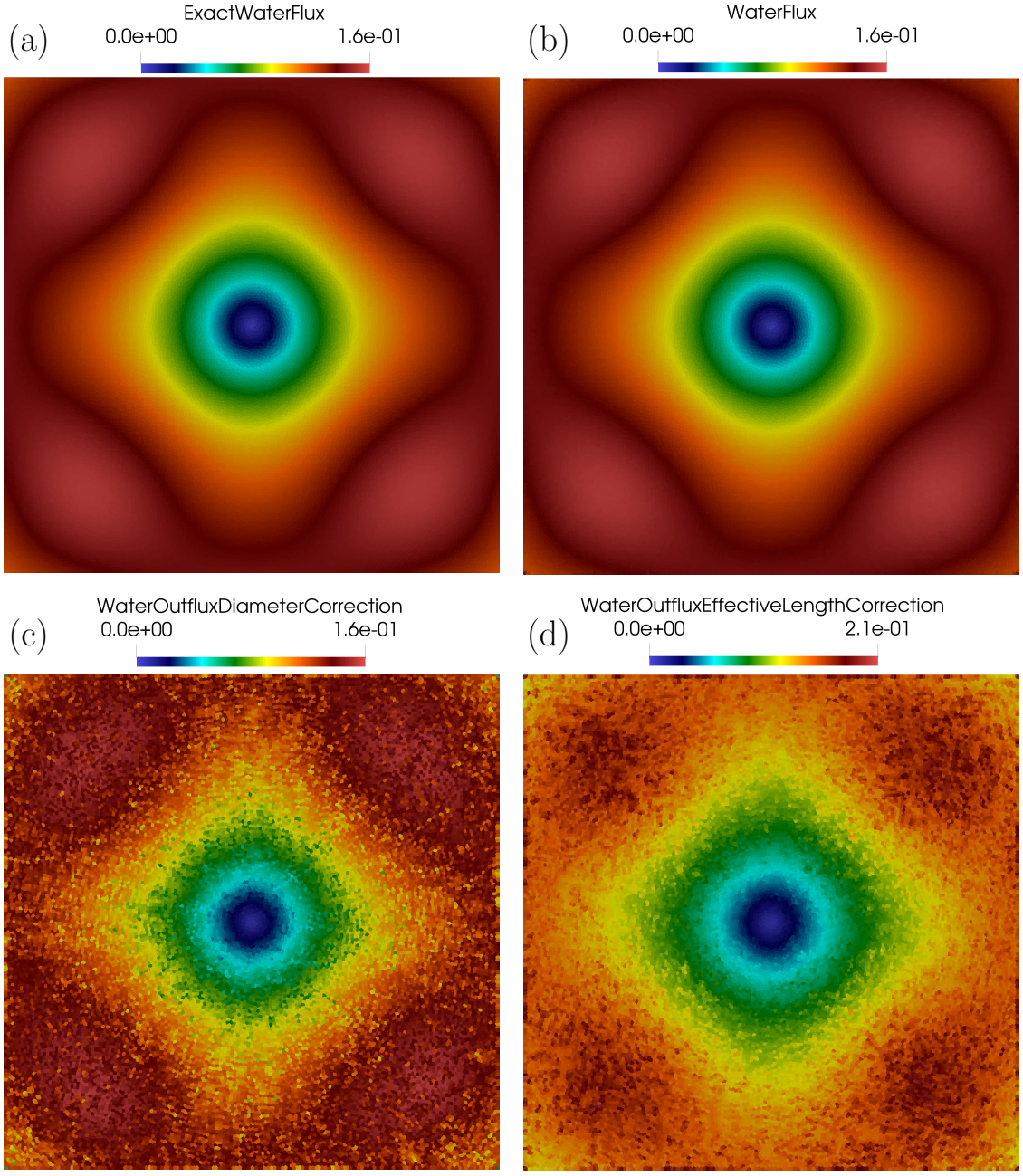



Figure 11: Water flux (top view) obtained for sinusoidal $k_m$ on the finest Voronoï mesh. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

appear, however $\widetilde{q}_K/w(K)$ is very noisy and influenced by the mesh cells shape, and in any case it is not a good

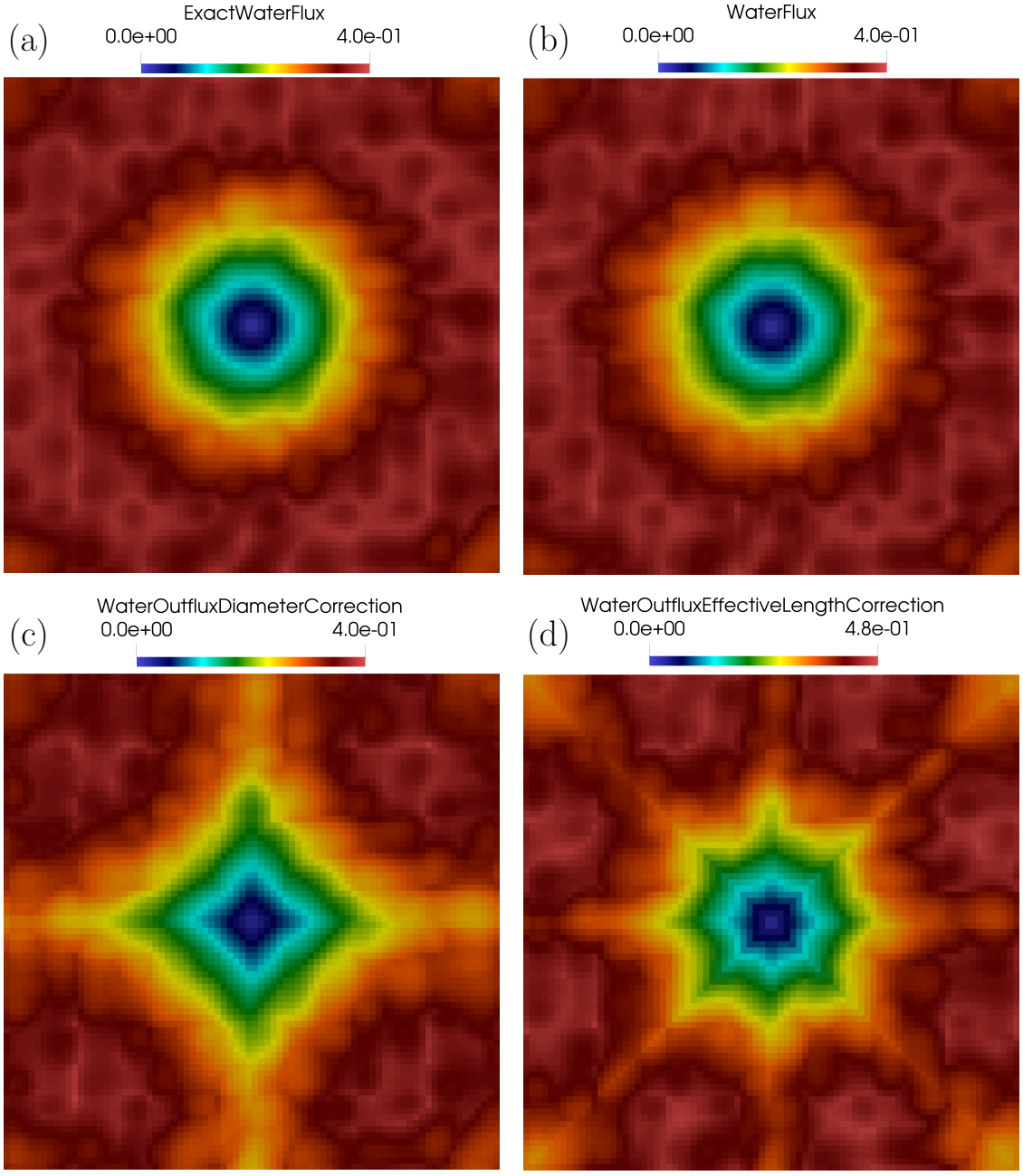



Figure 12: Water flux (top view) obtained for Perlin noised $k_m$ on the finest Cartesian mesh with square cells. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

approximation of $q_w$ or any continuous quantity. The observations are almost the same for the other choice of

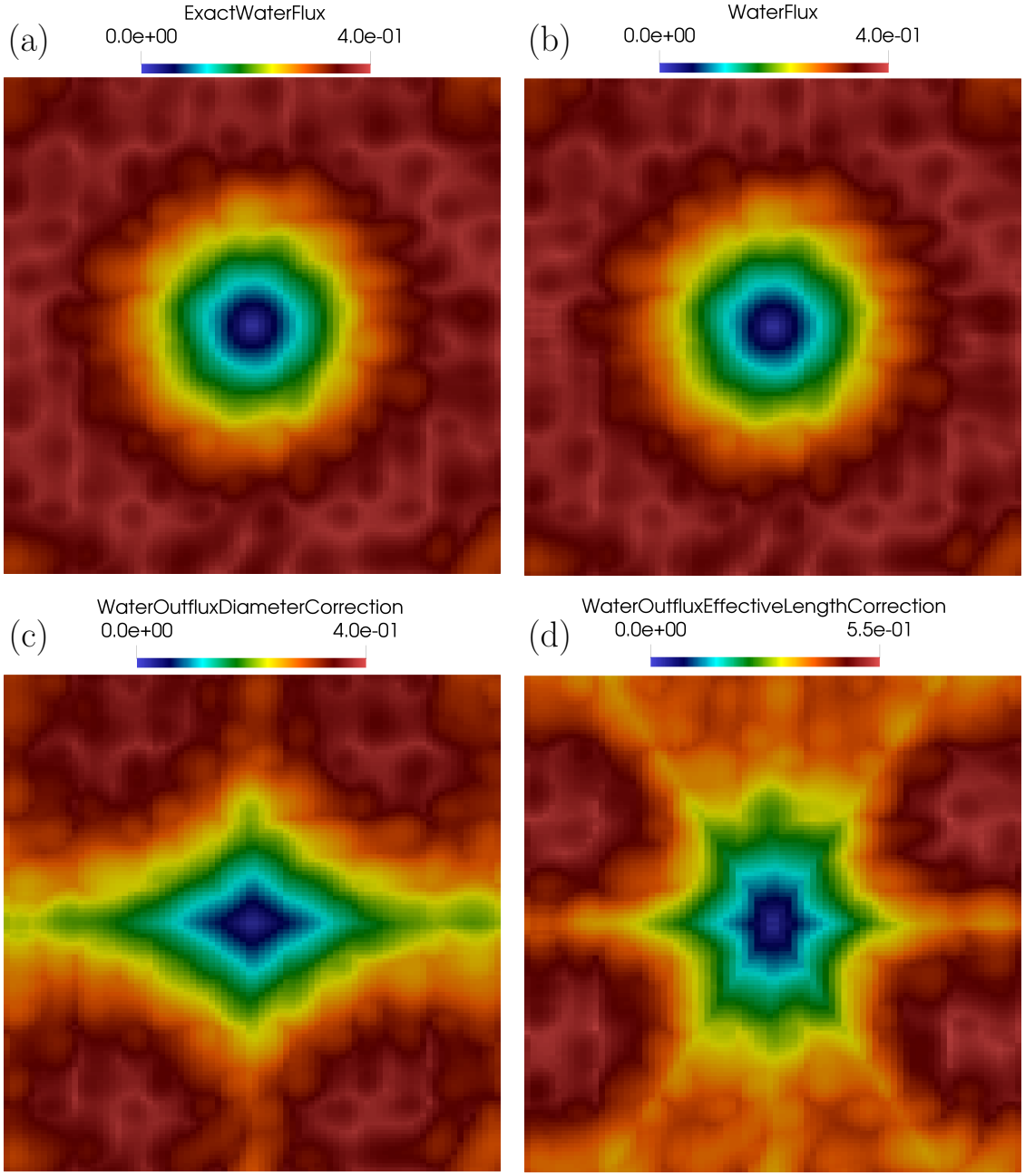



Figure 13: Water flux (top view) obtained for Perlin noised $k_m$ on the finest Cartesian mesh with rectangular cells. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

normalization where $w(K)$ is the effective length, with the main difference that one Cartesian meshes, $\widetilde{q}_K/w(K)$

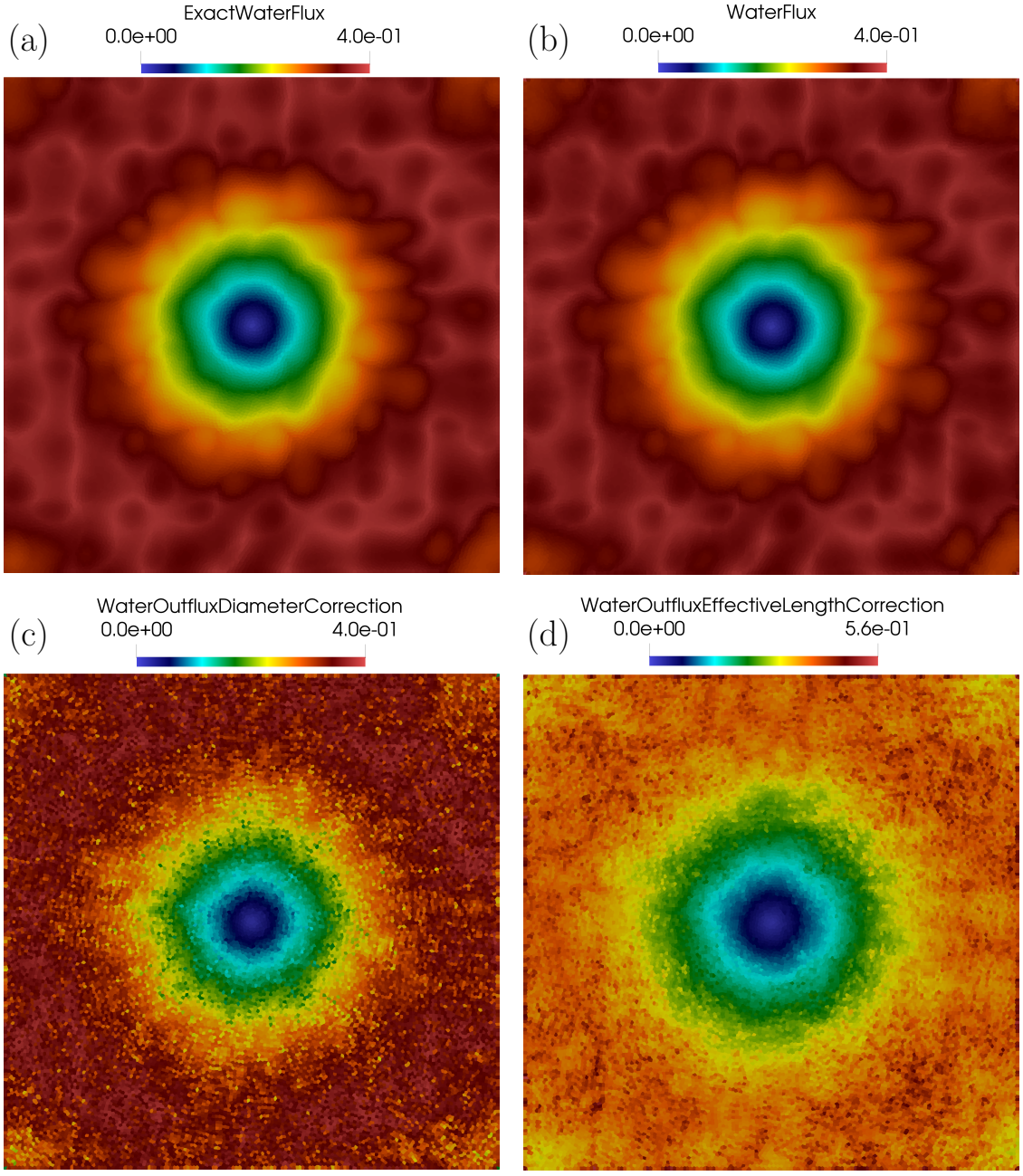



Figure 14: Water flux (top view) obtained for Perlin noised $k_m$ on the finest Voronoï mesh. (a) - Exact solution - (b) - Simulated water flux $q_K$ - (c) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the cell diameter - (d) - Simulated normalized water outflux $\widetilde{q}_K/w(K)$ with $w(K)$ equal to the effective flow length in the cell.

is much less biased towards the Cartesian directions. Those experiments confirm our theoretical observation

on $\widetilde{q}_K/w(K)$ and $q_K$: the former cannot be a good discrete approximation since it is strongly mesh dependent, while the later is a consistent approximation of $q_w$, which converges well to $q_w$ even on Voronoï meshes for the difficult case of the Perlin noised speed coefficient.

### 2.4.2 Second test configuration: perturbed mono-dimensional topography

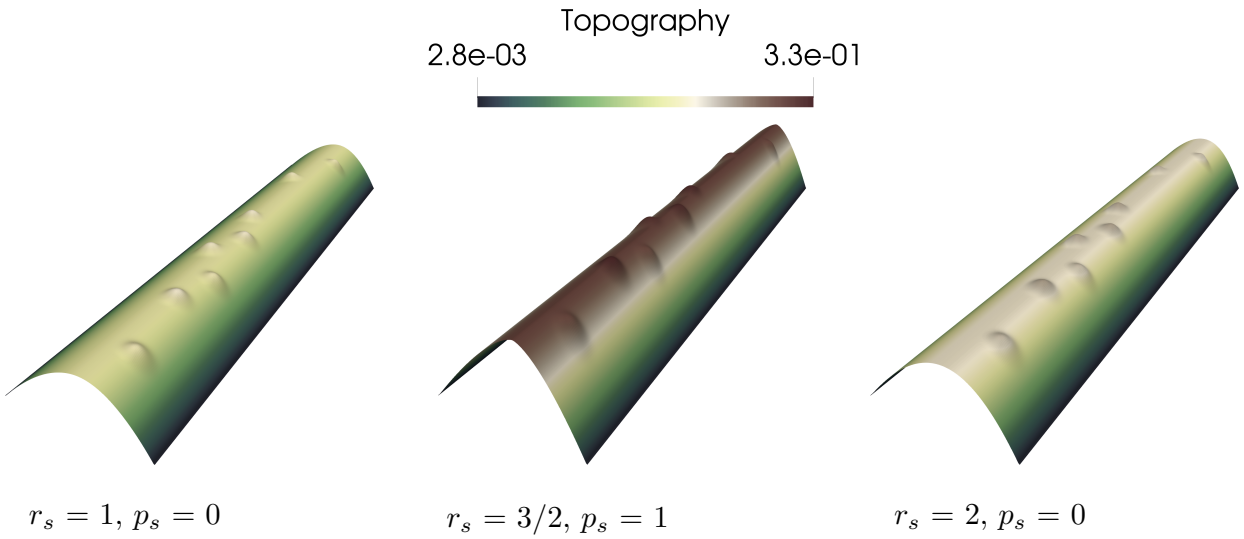



Figure 15: The three topographies $h_s$ of the second test configuration

The domain $\Omega$ is the rectangle $\Omega = ]0, L_x[\times]0, L_y[$ with $L_x = 1$ and $L_y = 5$, and the topography $h_s$ and water height $h_w^{ex}$ take the form:

$$\left| \begin{array}{l} h_s(x,y) = h_{s,x}(x) + \displaystyle\sum_{p=1}^{N_b} g_b\left(\frac{x-x_p}{\delta_x}, \frac{y-y_p}{\delta_y}\right), \\ h_w^{ex}(x,y) = h_{w,x}(x), \end{array} \right.$$

The topography $h_s$ incorporates $N_b$ small smooth bumps randomly positioned at points $(x_p, y_p)$ chosen such that they do not interfere with the boundaries of the domain, with the smooth bump function given by:

$$g_b(x,y) = g_b(r^2) = \left| \begin{array}{ll} H_{pert} \exp\left(\dfrac{-\gamma}{1-r^2}\right) \exp(\gamma) & \text{for } r^2 = x^2 + y^2 \leqslant 1, \\ 0 & \text{otherwise .} \end{array} \right.$$

In we use practice $N_b = 8$, $H_{pert} = 0.03$, $\gamma = 10$ and $\delta_x{=}\delta_y = 0.25$. For the mono-dimensional functions $(h_{s,x}, h_{w,x})$ we have chosen to re-use some of the stationary solutions proposed in [12]. Those functions are parametrized by six parameters $r_s$, $p_s$, $k_g$, $k_w$, $S_{s,x}$ and $S_{w,x}$. In the following test case, we will restrict ourselves to the configuration $k_g = k_w = 5$ as well as $S_{s,x} = 10$ and $S_{w,x} = 1$, while the couple $(r_s, p_s)$ will take the values $(1, 0)$, $(3/2, 1)$ and $(2, 0)$ to change the curvature of the topography. The corresponding mono-dimensional functions $h_{s,x}$ are such that $h_{s,x}(-x, y) = h_{s,x}(x, y)$, and they are given for $x \geqslant 0$ by:

. In the case $(r_s, p_s) = (1, 0)$:

$$h_{s,x} = h_{s,x}(0) - S_{s,x}\left(\frac{x}{k_w S_{w,x}} - \frac{k_g}{k_w^2 S_{w,x}^2}\ln|k_g + k_w S_{w,x} x| + \frac{k_g}{k_w^2 S_{w,x}^2}\ln|k_g|\right),$$

. In the case $(r_s, p_s) = (3/2, 1)$:

$$h_{s,x} = h_{s,x}(0) - \frac{p_s + 1}{p_s r_s}\frac{S_{s,x}^{\frac{1}{p_s+1}}}{k_w S_{w,x}^{r_s}}((k_g + k_w S_{w,x}^{r_s} x^{r_s})^{p_s/(p_s+1)} - k_g^{p_s/(p_s+1)})$$

. In the case $(r_s, p_s) = (2, 0)$:

$$h_{s,x} = h_{s,x}(0) - \frac{S_{s,x}}{2k_w S_{w,x}^2}(\ln\left(k_g + k_w S_{w,x}^2 x^2\right) - \ln\left(k_g\right)).$$

In all cases the value for $h_{s,x}(0)$ is such that $h_{s,x}(Lx/2) = 0$. To avoid some numerical truncation errors for higher values of $p_w$, for the water height we do not exactly follow [12] and use instead the simpler function:

$$h_{w,x} = (S_{w,x} x)(k_g + k_w (S_{w,x} x)^{r_s})^{\frac{p_w+1}{p_s+1}},$$

For this second configuration, we consider the three values 0, 1 and 2 for the slope exponent $p_w$. We restrict ourselves to the case of a constant speed coefficient $k_m = 1$ and the Cartesian mesh sequence with square cells.

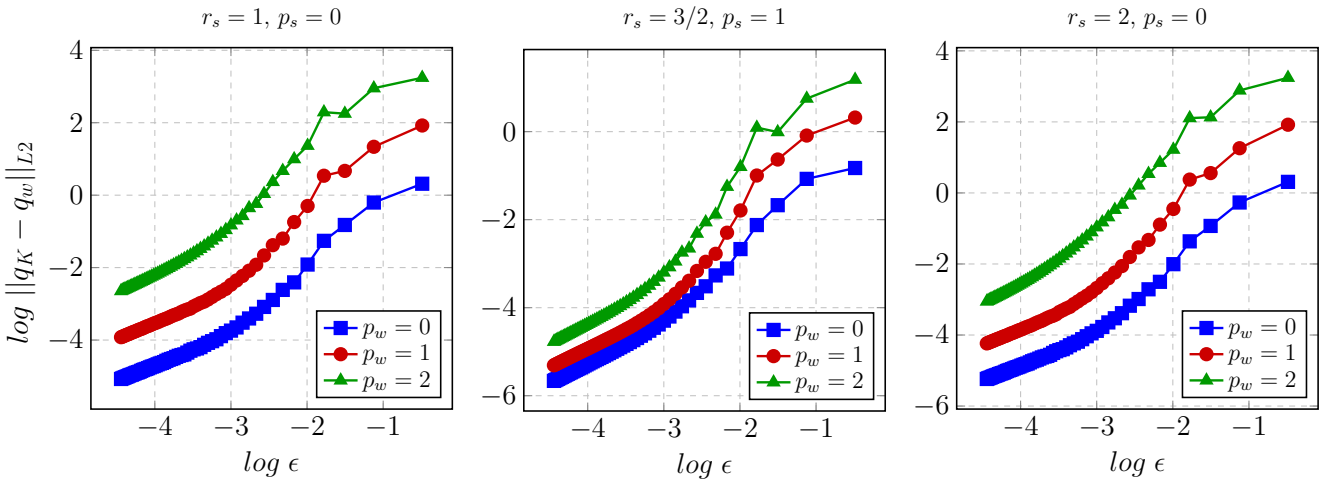



Figure 16: Convergence curves for $q_K$ towards $q_w$ for the second test configuration, for each value of $p_w$ and for each value of the couple $(r_s, p_s)$

On figure 16, we show the convergence curves for this second test configuration for each value of $p_w$ and each choice of the couple $(r_s, p_s)$. We clearly recover the expected convergence, confirming the ability of $q_k$ to approximate $q_w$ for various values of $p_w$.

# 3 A consistency post-processing for the classical node-to-node MFD/SFD algorithms

The objective of this section is to explain how to recover consistency by a simple post-processing step for the most classical node-to-node flow routing algorithms, again through establishing a link with a discretization of the continuous Gauckler-Manning-Strickler model. Such an explicit interpretation seems to be absent from the literature, so at least to the author's knowledge the results of this section are completely new. For simplicity we restrict ourselves in this section to uniform Cartesian meshes, and we adopt the usual Cartesian index $(i, j)$ notation for designating its nodes (see Fig. 17) as well as $\Delta x$ and $\Delta y$ for the Cartesian cell side lengths. This is by no means a restriction but simply a more convenient way to explain how to link node-to-node flow routing with Gauckler-Manning-Strickler models.

## 3.1 Node-to node MFD/SFD algorithms and the GMS model

In order to reinterpret the node-to-node flow routing algorithms as finite volume schemes, we must associate a volume to each node. The easiest way to do so is to consider the dual mesh, formed by joining the centers of the cells of the primal mesh (see again Fig. 17, where the dual mesh corresponds to the dashed lines). On the dual mesh, the node $(i, j)$ of the primal mesh becomes the center of the dual cell $K_{i,j}$.

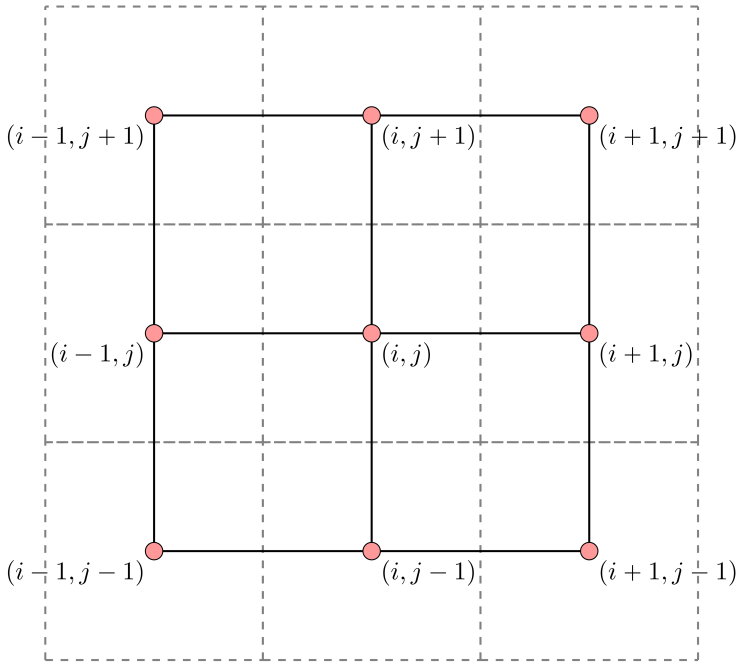



Figure 17: The Cartesian mesh (plain lines) and its dual (dashed lines)

In Fig. 18, we propose a decomposition of the boundary of the dual Cartesian cell $K_{i,j}$ centered on the primal internal node $(i, j)$ into 12 faces $(\sigma_l)_{1 \leqslant l \leqslant 12}$. The faces $\sigma_{j\pm1}$ are of length $\gamma_x \Delta_x$, and the faces $\sigma_{j\pm1}^{i\pm1}$ of length $\frac{1-\gamma_x}{2}\Delta x$. In the same way, faces $\sigma_{i\pm1}$ are of length $\gamma_y \Delta_y$ and the faces $\sigma_{i\pm1}^{j\pm1}$ of length $\frac{1-\gamma_y}{2}\Delta y$. Using those

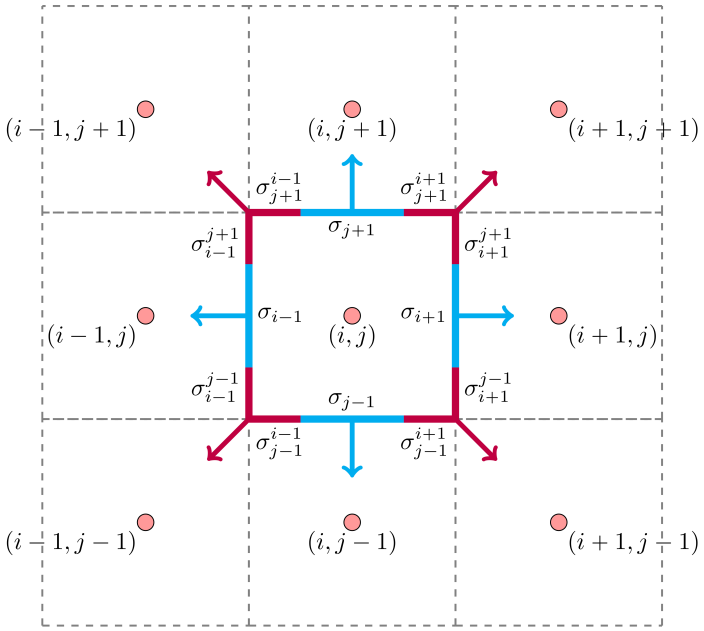



Figure 18: Decomposition and notations for the dual Cartesian cell boundaries

notations, we integrate (3) over the dual cell $K_{i,j}$ to get:

$$-\sum_{l=1}^{12}\int_{\sigma_l} k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n}_{K_{i,j}} = |K_{i,j}| S_{w,K_{i,j}}.$$

On the four faces $\sigma_{i-1}$, $\sigma_{i+1}$, $\sigma_{j-1}$ and $\sigma_{j+1}$, we use the same finite volume discretization than before:

$$\int_{\sigma_{j-1}} k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n}_{K_{i,j}} \approx \frac{\gamma_x \Delta x}{s_{ref}^{p_w} \Delta y} k_{m,\sigma_{j-1}} ||\boldsymbol{G}_{s,\sigma_{j-1}}||^{p_w} h_{w,\sigma_{j-1}}^{up} (h_{s,i,j-1} - h_{s,i,j}),$$

and

$$\int_{\sigma_{i-1}} k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n}_{K_{i,j}} \approx \frac{\gamma_y \Delta y}{s_{ref}^{p_w} \Delta x} k_{m,\sigma_{i-1}} ||\boldsymbol{G}_{s,\sigma_{i-1}}||^{p_w} h_{w,\sigma_{i-1}}^{up} (h_{s,i-1,j} - h_{s,i,j}),$$

and

$$\int_{\sigma_{j+1}} k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n}_{K_{i,j}} \approx \frac{\gamma_x \Delta x}{s_{ref}^{p_w} \Delta y} k_{m,\sigma_{j+1}} ||\boldsymbol{G}_{s,\sigma_{j+1}}||^{p_w} h_{w,\sigma_{j+1}}^{up} (h_{s,i,j+1} - h_{s,i,j}),$$

and

$$\int_{\sigma_{i+1}} k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n}_{K_{i,j}} \approx \frac{\gamma_y \Delta y}{s_{ref}^{p_w} \Delta x} k_{m,\sigma_{i+1}} ||\boldsymbol{G}_{s,\sigma_{i-1}}||^{p_w} h_{w,\sigma_{i+1}}^{up} (h_{s,i+1,j} - h_{s,i,j}),$$

while for the remaining height cells, we gather the faces to form the corners illustrated in Fig. (18). More precisely, we denote:

$$\overline{\sigma}_{i-1,j-1} = \overline{\sigma}_{i-1}^{j-1} \cup \overline{\sigma}_{j-1}^{i-1},$$

$$\overline{\sigma}_{i-1,j+1} = \overline{\sigma}_{i-1}^{j+1} \cup \overline{\sigma}_{j+1}^{i-1},$$
$$\overline{\sigma}_{i+1,j-1} = \overline{\sigma}_{i+1}^{j-1} \cup \overline{\sigma}_{j-1}^{i+1},$$
$$\overline{\sigma}_{i+1,j+1} = \overline{\sigma}_{i+1}^{j+1} \cup \overline{\sigma}_{j+1}^{i+1},$$

those four corners, $\sigma_{i\pm1,j\pm1}$ thus being the corner corresponding to the neighboring cell $K_{i\pm1,j\pm1}$. On those corners, we perform the same discretization than before considering the whole corner as if it was a single face: in other words we use constant values $k_{m,\sigma}$ and $||\boldsymbol{G}_{s,\sigma}||^{p_w}$ for $k_m$ and $||\nabla h_s||^{p_w}$ along the corner, an upwind scheme for the unknown $h_w\eta_w(h_w)$ and the a two-point flux formula for in the average normal direction to the corner. Denoting $(\nabla h_s)_{\sigma_{i\pm1,j\pm1}}$ the equivalent constant gradient exact for linear function underlying the TFPA along the corner, this leads to the following approximation:

$$\int_{\sigma_{i\pm1,j\pm1}} k_m h_w \eta_w(h_w) s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s \cdot \boldsymbol{n}_{K_{i,j}} \approx$$

$$k_{m,\sigma_{i\pm1,j\pm1}} s_{ref}^{-p_w} ||\boldsymbol{G}_{s,\sigma_{i\pm1,j\pm1}}||^{p_w} h_{w,\sigma_{i\pm1,j\pm1}}^{up} (\nabla h_s)_{\sigma_{i\pm1,j\pm1}} \cdot \int_{\sigma_{i\pm1,j\pm1}} \boldsymbol{n}_{K_{i,j}}.$$

where along each $\sigma_{i\pm1,j\pm1}$ we have again used a constant approximate value $k_{m,\sigma_{i\pm1,j\pm1}}$ and $\boldsymbol{G}_{s,\sigma_{i\pm1,j\pm1}}$ for $k_m$ and $||\nabla h_s||^{pw}$ along each face $\sigma$ and to use an upwind scheme $h_{w,\sigma_{i\pm1,j\pm1}}^{up}$ for the true unknown $h_w\eta_w(h_w)$. By construction, we have:

$$\int_{\sigma_{i\pm1,j\pm1}} \boldsymbol{n}_{K_{i,j}} = \pm\frac{(1-\gamma_y)}{2}\Delta y \boldsymbol{e_x} \pm \frac{(1-\gamma_x)}{2}\Delta x \boldsymbol{e_y}.$$

Denoting

$$|\sigma_{i\pm1,j\pm1}| = \frac{(1-\gamma_x)}{2}\Delta x + \frac{(1-\gamma_y)}{2}\Delta y = \delta,$$

we seek $\gamma_x$ and $\gamma_y$ such that:

$$\frac{(1-\gamma_x)}{2\delta}\Delta x = \frac{\Delta_y}{(\Delta_x^2+\Delta_y^2)^{1/2}} \quad \text{and} \quad \frac{(1-\gamma_x)}{2\delta}\Delta y = \frac{\Delta_x}{(\Delta_x^2+\Delta_y^2)^{1/2}},$$

leading to:

$$\gamma_x = 1 - \frac{2\delta\Delta_y/\Delta_x}{(\Delta_x^2+\Delta_y^2)^{1/2}} \quad \text{and } \gamma_y = 1 - \frac{2\delta\Delta_x/\Delta_y}{(\Delta_x^2+\Delta_y^2)^{1/2}}, \tag{15}$$

which can be achieved with $\gamma_x \geqslant 0$ and $\gamma_y \geqslant 0$ provided $\delta$ satisfies:

$$0 \leqslant \delta \leqslant \frac{1}{2}\min\left(\frac{\Delta x}{\Delta_y}, \frac{\Delta y}{\Delta_x}\right)(\Delta_x^2+\Delta_y^2)^{1/2}. \tag{16}$$

With this choice for $\gamma_x$ and $\gamma_y$, for all $\delta$ satisfying (16) we get that

$$\int_{\sigma_{i\pm1,j\pm1}} \boldsymbol{n}_{K_{i,j}} = \frac{\pm\Delta x}{(\Delta_x^2+\Delta_y^2)^{1/2}}\boldsymbol{e}_x + \frac{\pm\Delta y}{(\Delta_x^2+\Delta_y^2)^{1/2}}\boldsymbol{e}_y,$$

and thus the average normal at the corner $\sigma_{i\pm1,j\pm1}$ is precisely pointing from $\boldsymbol{x}_{K_{i,j}}$ to $\boldsymbol{x}_{K_{i\pm1,j\pm1}}$. Thus it is natural to use the two point flux formula:

$$(\nabla h_s)_{\sigma_{i\pm1,j\pm1}} \cdot \int_{\sigma_{i\pm1,j\pm1}} \boldsymbol{n}_{K_{i,j}} \approx \frac{\delta}{(\Delta_x^2+\Delta_y^2)^{1/2}}(h_{s,i\pm1,j\pm1} - h_{s,i}).$$

The upwinding is done exactly as before, following the sign of the difference in elevation $h_s + b$ between the two value forming the TPFA. This gives for the non-corners:

$$h^{up}_{w,\sigma_{i\pm1}} = \begin{cases} h_{w,i,j}\eta_w(h_{w,i,j}) & \text{if } h_{s,i,j} \geqslant h_{s,i\pm1,j}, \\ h_{w,i\pm1,j}\eta_w(h_{w,i\pm1,j}) & \text{if } h_{s,i,j} < h_{s,i\pm1,j}, \end{cases}$$

$$h^{up}_{w,\sigma_{j\pm1}} = \begin{cases} h_{w,i,j}\eta_w(h_{w,i,j}) & \text{if } h_{s,i,j} \geqslant h_{s,i,j\pm1}, \\ h_{w,i,j\pm1}\eta_w(h_{w,i,j\pm1}) & \text{if } h_{s,i,j} < h_{s,i,j\pm1}, \end{cases}$$

and for the corners:

$$h^{up}_{w,\sigma_{i\pm1,j\pm1}} = \begin{cases} h_{w,i,j}\eta_w(h_{w,i,j}) & \text{if } h_{s,i,j} \geqslant h_{s,i\pm1,j\pm1}, \\ h_{w,i\pm1,j\pm1}\eta_w(h_{w,i\pm1,j\pm1}) & \text{if } h_{s,i,j} < h_{s,i\pm1,j\pm1}. \end{cases}$$

For the gradient, we use for the non-corners:

$$\boldsymbol{G}_{s,\sigma_{i\pm1}} = \frac{1}{2}(\boldsymbol{G}_{s,K_{i\pm1,j}} + \boldsymbol{G}_{s,K_{i,j}}) \quad \text{and } \boldsymbol{G}_{s,\sigma_{j\pm1}} = \frac{1}{2}(\boldsymbol{G}_{s,K_{i,j\pm1}} + \boldsymbol{G}_{s,K_{i,j}})$$

and for the corners:

$$\boldsymbol{G}_{s,\sigma_{i\pm1,j\pm1}} = \frac{1}{2}(\boldsymbol{G}_{s,K_{i\pm1,j\pm1}} + \boldsymbol{G}_{s,K_{i,j}})$$

In the same way, for the face value of $k_m$ we use the following harmonic means for the non-corners:

$$k_{m,\sigma_{i\pm1}} = \frac{\overline{d}_{K_{i,j}K_{i\pm1,j}} k_{m,K_{i,j}} k_{m,K_{i\pm1,j}}}{k_{m,K_{i,j}}\overline{d}_{K_{i\pm1,j},\sigma_{i\pm1}} + k_{m,K_{i\pm1,j}}\overline{d}_{K_{i,j},\sigma_{i\pm1}}} = \frac{2k_{m,K_{i,j}}k_{m,K_{i\pm1,j}}}{k_{m,K_{i,j}} + k_{m,K_{i\pm1,j}}}$$

and

$$k_{m,\sigma_{j\pm1}} = \frac{\overline{d}_{K_{i,j}K_{i,j\pm1}} k_{m,K_{i,j}} k_{m,K_{i,j\pm1}}}{k_{m,K_{i,j}}\overline{d}_{K_{i,j\pm1},\sigma_{j\pm1}} + k_{m,K_{i,j\pm1}}\overline{d}_{K_{i,j},\sigma_{j\pm1}}} = \frac{2k_{m,K_{i,j}}k_{m,K_{i,j\pm1}}}{k_{m,K_{i,j}} + k_{m,K_{i,j\pm1}}}$$

and for the corners:

$$k_{m,\sigma_{i\pm1,j\pm1}} = \frac{\overline{d}_{K_{i,j}K_{i\pm1,j\pm1}} k_{m,K_{i,j}} k_{m,K_{i\pm1,j\pm1}}}{k_{m,K_{i,j}}\overline{d}_{K_{i\pm1,j\pm1},\sigma_{i\pm1,j\pm1}} + k_{m,K_{i\pm1,j\pm1}}\overline{d}_{K_{i,j},\sigma_{i\pm1,j\pm1}}} = \frac{2k_{m,K_{i,j}}k_{m,K_{i\pm1,j\pm1}}}{k_{m,K_{i,j}} + k_{m,K_{i\pm1,j\pm1}}}$$

where $\overline{d}_{K_{i,j},\sigma_{i\pm1,j\pm1}} = \sqrt{\Delta_x^2 + \Delta_y^2}$ is the distance between $K_{i,j}$ and the corner point belonging to $\sigma_{i\pm1,j\pm1}$. To get more compact notations, let us denote

$$\mathcal{N}(i,j) = \{(m,n) \in \{i-1, i, i+1\} \times \{j-1, j, j+1\} \mid (m,n) \leqslant (i,j)\},$$

the neighbors of node $(i,j)$, and define the transmissivities:

$$\tau^{m,n}_{i,j} = \begin{cases} \dfrac{\gamma_x \Delta x}{s^{p_w}_{ref}\Delta y} k_{m,\sigma_{j\pm1}} ||\boldsymbol{G}_{s,\sigma_{j\pm1}}||^{p_w} & \text{if } (m,n) = (i,j-1) \text{ or } (i,j+1), \\ \dfrac{\gamma_y \Delta y}{s^{p_w}_{ref}\Delta x} k_{m,\sigma_{i\pm1}} ||\boldsymbol{G}_{s,\sigma_{i\pm1}}||^{p_w} & \text{if } (m,n) = (i-1,j) \text{ or } (i+1,j), \\ \dfrac{\delta}{s^{p_w}_{ref}(\Delta_x^2 + \Delta_y^2)^{1/2}} k_{m,\sigma_{i\pm1,j\pm1}} ||\boldsymbol{G}_{s,\sigma_{i\pm1,j\pm1}}||^{p_w} & \text{otherwise,} \end{cases}$$

assuming for simplicity that the gradients $\boldsymbol{G}_{s,\sigma}$ are obtained on the dual mesh in the same way as in the cell-to-cell case (of course, a reconstruction formula using also the diagonal neighbors is possible). Using those

notations, we get gathering by upwind kind as in the case of the cell-to-cell flow routing the following expression for the proposed finite volume scheme on the dual mesh:

$$h_{w,i,j}\eta_w(h_{w,i,j})\left(\sum_{(m,n)\in\mathcal{N}(i,j),h_{s,i,j}>h_{s,m,n}}\tau_{i,j}^{m,n}(h_{s,i,j}-h_{s,m,n})\right)$$

$$-\left(\sum_{(m,n)\in\mathcal{N}(i,j),h_{s,i,j}<h_{s,m,n}}\tau_{i,j}^{m,n}h_{w,m,n}\eta_w(h_{w,m,n})(h_{s,m,n}-h_{s,i,j})\right)=|K_{i,j}|S_{w,i,j}.$$

Proceeding as in the cell-to-cell case, denoting:

$$s_{i,j}=\sum_{(m,n)\in\mathcal{N}(i,j),h_{s,i,j}>h_{s,m,n}}\tau_{i,j}^{m,n}(h_{s,i,j}-h_{s,m,n})\quad\text{and}\quad\widetilde{q}_{i,j}=h_{w,i,j}\eta_w(h_{w,i,j})s_{i,j},$$

we finally get:

$$\widetilde{q}_{i,j}-\left(\sum_{(m,n)\in\mathcal{N}(i,j),h_{s,i,j}<h_{s,m,n}}\tau_{i,j}^{m,n}\frac{\widetilde{q}_{m,n}}{s_{m,n}}(h_{s,m,n}-h_{s,i,j})\right)=|K_{i,j}|S_{w,i,j}. \tag{17}$$

The flow sharing formula common to all flow routing algorithms of the literature identifies in this context with the ratios:

$$\frac{1}{s_{m,n}}\tau_{i,j}^{m,n}(h_{s,m,n}-h_{s,i,j}),$$

for $(m,n)\in\mathcal{N}(i,j), h_{s,i,j}<h_{s,m,n}$, which expresses how node $(i,j)$ receives water from other nodes. Reversing the point of view, it rewrites in probably more familiar fashion by expressing how node $(i,j)$ distributes water to its neighbors through the flow sharing formula (noticing that $\tau_{i,j}^{m,n}=\tau_{m,n}^{i,j}$):

$$\frac{\tau_{i,j}^{m,n}\max(0,h_{s,i,j}-h_{s,m,n})}{\sum_{m',n'\in\mathcal{N}(i,j)}\tau_{i,j}^{m',n'}\max(0,h_{s,i,j}-h_{s,m',n'})}. \tag{18}$$

Notice that several attempts of the literature at improving the behavior of the flow routing consider powers $q$ of the two point slope instead of the slope in the flow sharing formula, which with our notations rewrites:

$$\frac{\tau_{i,j}^{m,n}\max(0,h_{s,i,j}-h_{s,m,n})^q}{\sum_{m',n'\in\mathcal{N}(i,j)}\tau_{i,j}^{m',n'}\max(0,h_{s,i,j}-h_{s,m',n'})^q}. \tag{19}$$

Another important consequence of the formal identification of cell-to-cell flow routing algorithms with a numerical scheme for the stationary Gauckler-Manning-Strickler model is the fact that if one wants to incorporate powers of the slope in the flow distribution procedure, then one should not use powers of the directional slope $\frac{1}{\overline{d}_{KL}}(h_{s,L}-h_{s,K})$ but rather use powers of $||\boldsymbol{G}_{s,\sigma}||$ to remain consistent with a continuous model incorporating powers of $||\nabla h_s||$. Otherwise, the consistency of the flow routing algorithm will be lost again. In [42] it is even suggested to choose different values of $q$ for different grid sizes, emphasizing this non-consistency. However, the sought flow concentration effect can be achieved in a consistent manner by (18) through the use of $p_w$ with value $p_w=q-1$: the full gradient and not only the directional gradient being used this way, this does not compromise consistency and a value independent of the mesh should be chosen according to physical considerations. An option that we do not consider here is to make the value of $p_w$ spatially variable, as was suggested in [40] but still on the non-consistent formulation (19).

Although (19) clearly leads to some non consistency, this expression is useful to derive a classification of the most prominent flow routing algorithms of the literature. To exactly match the definitions of most node-to-node flow routing schemes of the literature, we now consider the special case of square cartesian cells for which $\Delta_x = \Delta_y = \Delta_{xy}$. In this case we get from (15) that $\gamma_x = \gamma_y = 1 - (2\delta)/(\sqrt{2}\Delta_{xy})$. It remains to choose a value for $\delta$. The most natural choice is choose to enforce $\delta = \gamma_x \Delta_x = \gamma_y \Delta y$ and thus balance the contribution to each neighbor. This immediately leads to:

$$\delta = \frac{\sqrt{2}}{2+\sqrt{2}}\Delta_{xy} \quad \text{and} \quad \gamma_x = \gamma_y = \frac{\sqrt{2}}{2+\sqrt{2}},$$

implying that:

$$\frac{\delta}{(\Delta_x^2+\Delta_y^2)^{1/2}} = \frac{1}{2+\sqrt{2}} \quad \text{and} \quad \frac{\gamma_x \Delta_x}{\Delta_y} = \frac{\sqrt{2}}{2+\sqrt{2}} \quad \text{and} \quad \frac{\gamma_y \Delta_y}{\Delta_x} = \frac{\sqrt{2}}{2+\sqrt{2}},$$

thus the diagonal transmissivities differ from the non-diagonal ones by the factor $1/\sqrt{2}$ which corresponds to the D8, Rho8 and most MFD algorithms. To recover the FD8/TOPMODEL noticing that the $L_1$ and $L_2$ non diagonal and diagonal "face measures" of this MFD algorithm satisfy $L_1 = \Delta_{xy}/2$ and $L_2 = \frac{\sqrt{2}}{4}\Delta_{xy}$, we recover the same weighting within our notations by setting

$$\delta = \frac{\sqrt{2}}{4}\Delta_{xy} \quad \text{and} \quad \gamma_x = \gamma_y = \frac{1}{2},$$

which is compatible with (15) as in this case:

$$\frac{(1-\gamma_x)}{2\delta}\Delta_x = \frac{(1-\gamma_y)}{2\delta}\Delta_y = \frac{1}{\sqrt{2}} = \frac{\Delta_x}{(\Delta_x^2+\Delta_y^2)^{1/2}} = \frac{\Delta_y}{(\Delta_x^2+\Delta_y^2)^{1/2}}.$$

Finally denoting:

$$\Delta\mathcal{H}_{i,j}^{m,n} = \max(0, h_{s,i,j} - h_{s,m,n}),$$

in table 1 we recast the most classical MFD algorithms using our notations, with $p_w = 0$ for all the presented methods. For the Rho8 method ([20]), the $\rho_8$ parameter is a random number generated for each face, while for the MFD-md ([40]), the parameter $e$ is the maximum downslope gradient and $f(e) = 8.9\min(e, 1) + 1.1$.

We see from table 1 that many classical MFD/SFD schemes can be reinterpreted in the context of the GMS model. In particular, the MFD (Freeman 1989) and the FD8 algorithms exactly correspond to the quantity $\widetilde{q}_K$ obtained from a coherent discretization of the GMS model, since they set $q = 1$. Meanwhile, notice that MFD (Freeman 1991) and MFD (Holmgren 1994) are attempts to correct $\widetilde{q}_{i,j}$ obtained given by (Freeman 1989) by using a value of $q$ not equal to one, and the same holds for TOPMODEL and MFD-md with respect to FD8. For those scheme, before reconstructing $q_{i,j}$ from $\widetilde{q}_{i,j}$ as we will detail in next subsection, one must first go back to the correct value $q = 1$ to recover a consistent approximation of $q_w$. The sought flow concentration effect of those schemes can be recovered in a consistent fashion by increasing the value of $p_w$ instead of introducing $q$.

Finally, it is clear that the chosen value for $k_{m,\sigma}$ should be a discretization of an inverse of a continuous roughness with a more or less physical interpretation. Apart from the unavoidable sampling induced by the mesh, it should be as mesh independent as possible and in particular should not depend on cell orientations. The single flow direction D8 and Rho8 methods reinterpreted this way introduce a coefficient $k_{m,\sigma}$ that is clearly mesh dependent and not the discretization of a continuous coefficient. This will consequently increase the mesh dependency of the overall method. Again, to recover the flow concentration/monodirectional effect sought by SFD algorithm, on should instead resort to a continuous speed coefficient map $k_m$ that presents high contrasts with narrow zone with high values. The flow will then preferentially follow those regions creating the desired

Table 1: A possible classification of MFD algorithms using (19)

| Method | $\delta/\Delta_{xy}$ | $\gamma_x = \gamma_y$ | $q$ | $k_{m,\sigma_{m,n}}$ |
|---|---|---|---|---|
| D8 (O'Callaghan et al. 1984 [33]) | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | 1 | $\begin{cases} 1 & \sigma_{m,n} \text{ has largest } \Delta\mathcal{H}_{i,j}^{m,n} \\ 0 & \text{otherwise} \end{cases}$ |
| MFD (Freeman 1989 [22]) | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | 1 | 1 |
| MFD (Freeman 1991 [23]) | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | 1.1 | 1 |
| Rho8 (Fairfield 1991 [20]) | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | 1 | $\begin{cases} 1 & \sigma_{m,n} \text{ has largest } \rho_8\Delta\mathcal{H}_{i,j}^{m,n} \\ 0 & \text{otherwise} \end{cases}$ |
| FD8 (Quinn et al. 1991 [41]) | $\frac{\sqrt{2}}{4}$ | $\frac{1}{2}$ | 1 | 1 |
| MFD (Holmgren 1994 [28]) | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | $\in [1, \infty[$ | 1 |
| TOPMODEL (Quinn et al. 1995 [42]) | $\frac{\sqrt{2}}{4}$ | $\frac{1}{2}$ | $\in [1 - 100]$ | 1 |
| MFD-md (Qin et al. 2007 [40]) | $\frac{\sqrt{2}}{4}$ | $\frac{1}{2}$ | $f(e)$ | 1 |

channelization effect (see [12]).

The two point flux approximation (TPFA) is of course not the only possible approximation for the terms $(\nabla h_s)_{\sigma_{i\pm1,j\pm1}} \cdot \int_{\sigma_{i\pm1,j\pm1}} \boldsymbol{n}$. In particular, if one reconstructs an approximation $\hat{\boldsymbol{G}}_{s,\sigma}$ of the full topographic gradient along each face $\sigma$, then it can be used to compute an approximation of the flux. We denote it $\hat{\boldsymbol{G}}_{s,\sigma}$ to distinguish it from the reconstruction $\boldsymbol{G}_{s,\sigma}$ used to approximate the non-linear dependency in the slope, as the two can be different. In this case, (19) becomes:

$$\frac{|\sigma_{m,n}|(\Delta\mathcal{H}_{i,j}^{m,n})^q}{\sum\limits_{m',n' \in \mathcal{N}(i,j)} |\sigma_{m',n'}|(\Delta\mathcal{H}_{i,j}^{m',n'})^q} \quad \text{and} \quad \Delta\mathcal{H}_{i,j}^{m,n} = \max\left(0, \hat{\boldsymbol{G}}_{s,\sigma_{m,n}} \cdot \int_{\sigma_{m,n}} \boldsymbol{n}_{K_{i,j}}\right). \tag{20}$$

Then, more flow routing algorithms of the literature can be rewritten this way. In particular, choosing $\gamma_x = \gamma_y = 0$ or 1 we can easily recover the flux decomposition method (Desmet et al. 1996 [16]) and a variation of the MD$\infty$ method (Seibert et al. 2007 [47]). The flux decomposition method chooses a single value for $\hat{\boldsymbol{G}}_{s,K_{i,j}}$ for each cell, and then loop over cells and set $\hat{\boldsymbol{G}}_{s,\sigma} = \hat{\boldsymbol{G}}_{s,K_{i,j}}$ for the faces of the current cell that have not already been handled through a previous cell in the loop. The MD$\infty$ methods computes $\hat{\boldsymbol{G}}_{s,\sigma}$ for each face using a triangular reconstruction of the slope: to be precise, with our notations $\hat{\boldsymbol{G}}_{s,\sigma}$ is for face $\sigma_{m,n}$ half the sum of the two triangular gradients computed in [47] that can contribute to $\sigma_{m,n}$. We refer to this as a variation of [47] as it is unclear whether they use the normal component of the gradient as we do here or the full norm of the gradient in their flow sharing formula.

Other flow routing algorithms that do not easily fit into this mathematical framework are also available in the literature. We mention in particular the ANSWERS ([6]), DEMON ([13]) and Lea's method ([29]), that are all based on a local planar approximation of the topography and use either a multiple or single direction flow sharing formula based on purely geometric considerations. The D$\infty$ method (Tarboton 1997 [48]) strongly looks

Table 2: A classification of some flow routing algorithms using (20)

| Method | $\delta/\Delta_{xy}$ | $\gamma_x = \gamma_y$ | $q$ | $k_{m,\sigma_l}$ |
|---|---|---|---|---|
| Flux decomposition (Desmet et al. 1996 [16]) | 0 | 1 | 1 | 1 |
| MD$\infty$ (Seibert et al. 2007, [47]) | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | $\frac{\sqrt{2}}{2+\sqrt{2}}$ | 1 | 1 |

like the SFD method at first sight, however because the flow sharing formula used when the steepest direction is not aligned with mesh direction is based on angular considerations similar to those of ANSWERS and DEMON, it is not immediately obvious how to relate the D$\infty$ method to a continuous model. Finally, let us mention that many variations around the classical algorithms have been explored since their first publications leading for instance to some generalization to triangular meshes [3, 57]. We refer the reader to [19, 56, 34] and references therein for a broader review on flow routing algorithms and their numerical behavior.

## 3.2 Consistency post-processing for node-to-node MFD algorithms

The node-to-node situation is no better than the cell-to-cell one: $\widetilde{q}_{i,j}$ will be as non consistent, non convergent and thus strongly mesh dependent than its cell-to-cell counterpart. The node-to-node routing is indeed simply a cell-to-cell routing on a dual mesh, with a more involved cell boundary decomposition. Again, the quantity $\widetilde{q}_{i,j}$ should not be used to couple with sediment evolution, one should instead reconstruct a consistent water flux vector $\boldsymbol{Q}_{i,j}$ for instance by setting:

$$\boldsymbol{Q}_{i,j} = \sum_{(m,n)\in\mathcal{N}(i,j),h_{s,i,j}>h_{s,m,n}} \frac{\tau_{i,j}^{m,n}\widetilde{q}_{i,j}}{|K_{i,j}|s_{i,j}}(h_{s,i,j}-h_{s,m,n})(\boldsymbol{x}_{i,j}^{m,n}-\boldsymbol{x}_{K_{i,j}})- \sum_{(m,n)\in\mathcal{N}(i,j),h_{s,i,j}<h_{s,m,n}} \frac{\tau_{i,j}^{m,n}\widetilde{q}_{m,n}}{|K_{i,j}|s_{m,n}}(h_{s,m,n}-h_{s,i,j})(\boldsymbol{x}_{i,j}^{m,n}-\boldsymbol{x}_{K_{i,j}}) \tag{21}$$

where:

$$\boldsymbol{x}_{i,j}^{m,n} = \begin{cases} \frac{1}{2}(\boldsymbol{x}_{K_{i,j}}+\boldsymbol{x}_{K_{m,n}}) & \text{if } (m,n)\in\{(i,j-1),(i,j+1),(i-1,j),(i+1,j)\} \\ \frac{1}{|\sigma_m^n|+|\sigma_n^m|}\left(|\sigma_m^n|\boldsymbol{x}_{\sigma_m^n}+|\sigma_n^m|\boldsymbol{x}_{\sigma_n^m}\right) & \text{otherwise} \end{cases}$$

and then use

$$q_{i,j} = ||\boldsymbol{Q}_{i,j}|| \tag{22}$$

which again can be considered as an easy to implement post-processing consistency correction step. Again, this should be done in conjunction with $q = 1$ and a mesh independent $k_m$ map.

## 3.3 Boundary terms

For the sake of completeness, let us briefly and graphically explain how to handle boundary terms. One first needs to distinguish between corner boundary points and other boundary points. For a corner boundary point $(i, j)$, we use the decomposition of the boundary of the dual cell presented on figure 19 while for a non corner boundary point $(i, j)$ we use the decomposition of the boundary of the dual cell presented on figure 20. The principle is exactly the same as for an interior node $(i, j)$, but with a reduced set of neighbors and with a

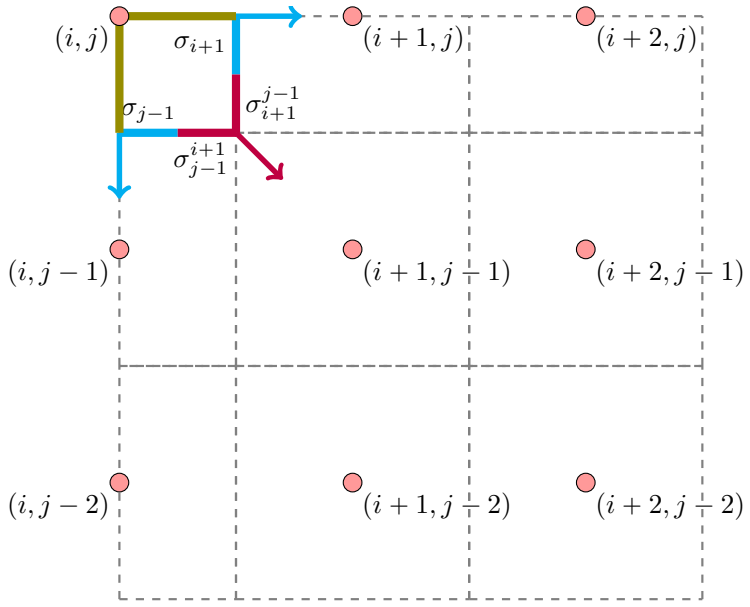



Figure 19: Decomposition and notations for the dual corner cartesian cell boundaries

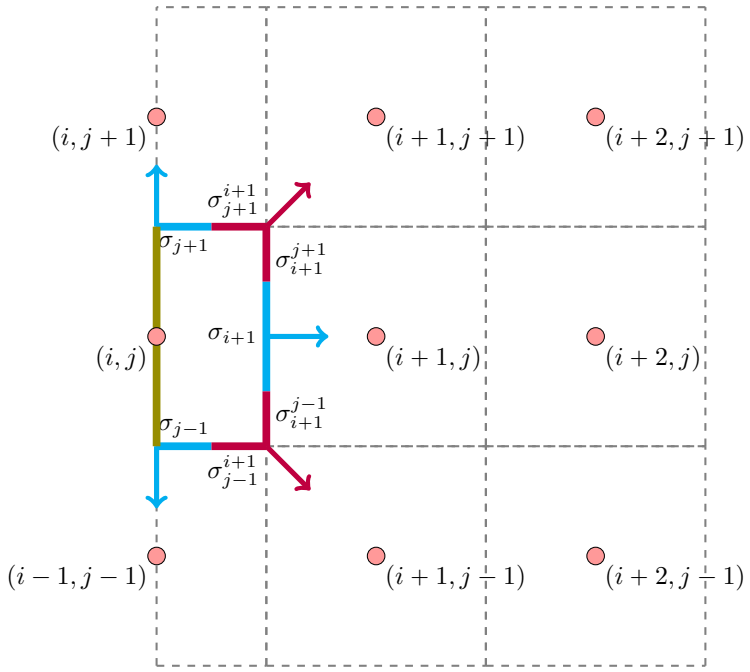



Figure 20: Decomposition and notations for the non-corner cartesian cell boundaries

half reduction of the face length and thus of $\tau_{i,j}^{m,n}$ for non-diagonal neighbors. For the source terms associated with those cells, one must also reduce by one fourth the magnitude of the cell size for a corner boundary point

and on half for a non corner boundary point, and those sources should be taken at the center of mass of the corresponding boundary dual cells. Finally, one needs to add to the source term the integral of $B_w$ along the part of the boundary of the cells that belong to $\partial\Omega_{in}$.

### 3.4 Numerical results for the corrected node-to-node MFD algorithms

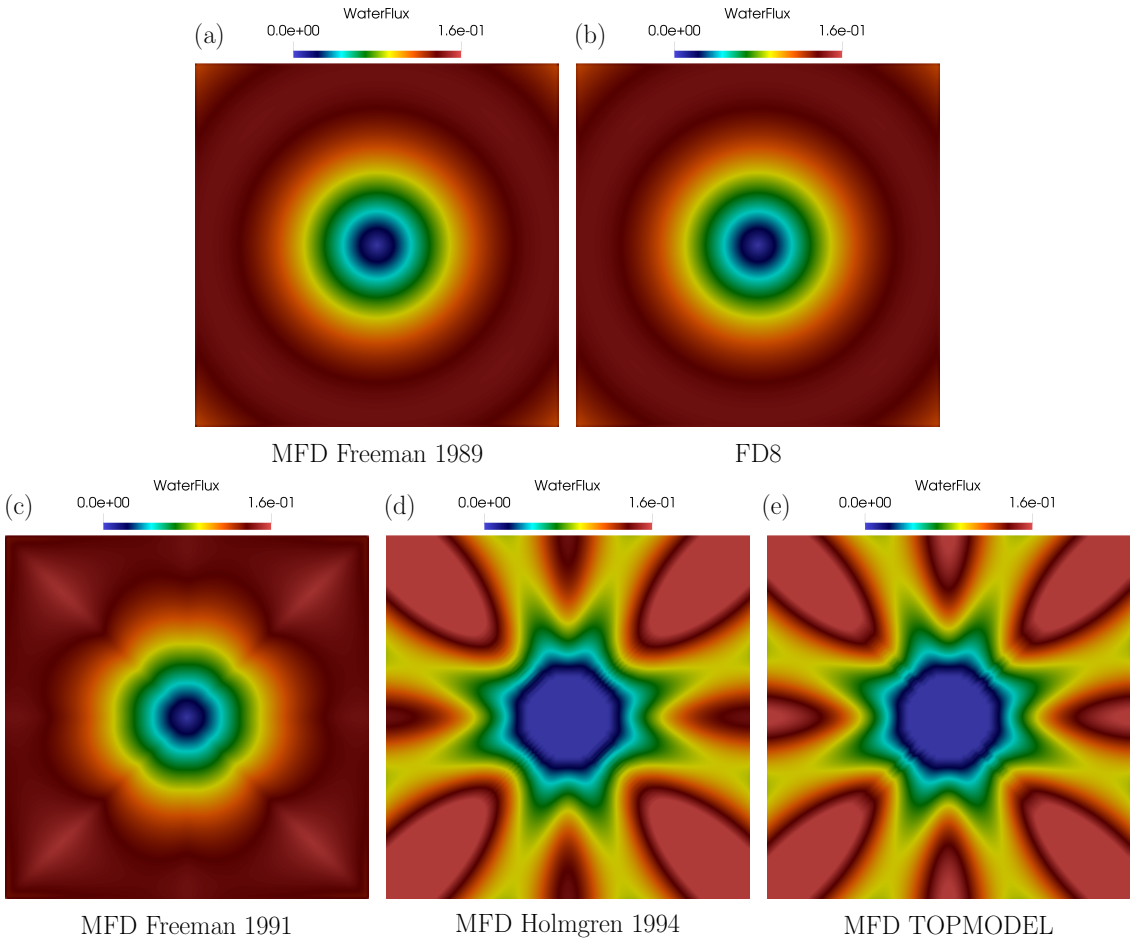



Figure 21: Results for constant $k_m$ on the finest Cartesian mesh with square cells sequence for the node-to-node MFD algorithms

We consider the same two test configurations than in the cell-to-cell case, and only the Cartesian mesh sequence with square cells. For the first test configuration, we start by presenting on figure 21 in the case of constant $k_m$ the reconstruction $q_{ij}$ for five MFD schemes: MFD (Freeman 1989), FD8, MFD (Freeman 1991), MFD (Holmgren 1994) and TOPMODEL, chossing a value $q = 10$ for MFD (Holmgren 1994) and TOPMODEL. As expected, the two schemes with $q = 1$, namely MFD (Freeman 1989) and FD8 lead to the correct solution. For the three others, because $q \neq 1$ we see that we do not get the expected result, thus confirming our theoretical observations. Of course, the same can be observed for the sinusoidal $k_m$ or the Perlin noised $k_m$. Since MFD (Freeman 1991), MFD (Holmgren 1994) are identical to MFD (Freeman 1989) and TOPMODEL and MFD-md are identical to FD8 if we take $q = 1$, for our remaining experiments we restrict ourselves to MFD (Freeman

1989) and FD8.

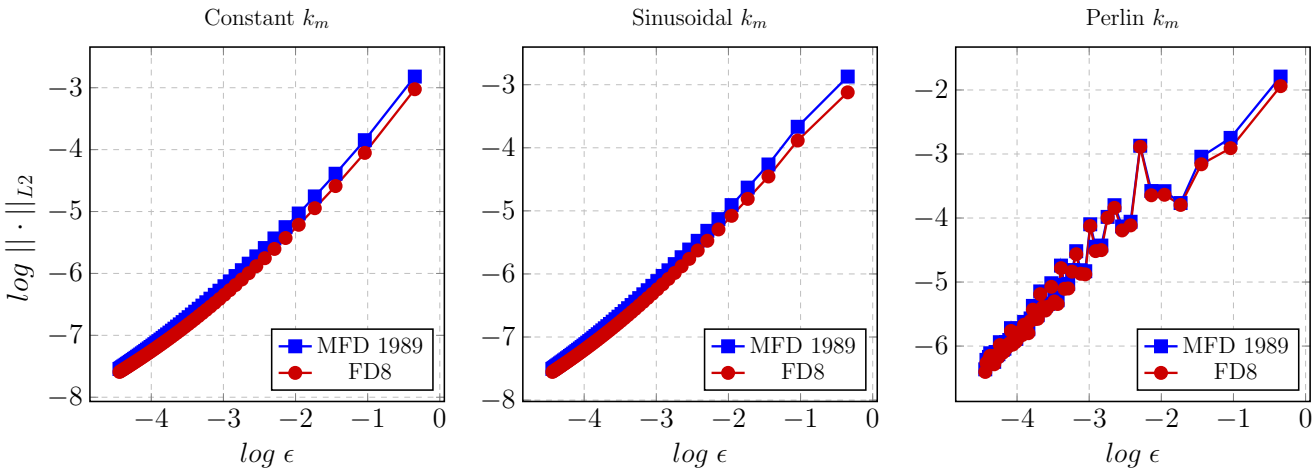



Figure 22: Convergence curves for $q_{i,j}$ towards $q_w$ for the first test configuration, for each choice for $k_m$

On figure 22, we present the convergence curves for those two schemes on the first test configuration for the three choices of speed coefficient $k_m$. We clearly recover the expected convergence, confirming the quality of the post-processing $q_{i,j}$ for node-to-node algorithms.
On figure 23, we present the convergence curves for those two schemes on the second test configuration for the three values of $p_w$ and the three choices of couple $(r_s, p_s)$. Again, those curves confirm the good behavior of $q_{i,j}$. Notice that the convergence curves are almost impossible to distinguish for this second test case configuration between the two schemes, indicating that they produce almost identical results.

## 4 Limitations of the GMS model

The application domain of the GMS model is limited by some additional requirements on the topography $h_s$. Because of the equivalence of MFD algorithms with the GMS model, they will suffer from the same limitations. Mathematically, systems (1) and (3) are stationary transport problems for $a$ or $h_w$. Their well-posedness, i.e. existence and uniqueness in a suitable function space and continuity with respect to data, is rigorously established only under some condition on the topography, all introducing some positivity requirement in the zero order part of the differential operators applied to $a$ or $h_w$ (see [11, 4, 52, 17, 21, 25]). In particular, among the possible assumptions on the topography the simplest ones are undoubtedly:

$$-\Delta h_s > 0 \quad \text{or} \quad -div\left(k_m s_{ref}^{-p_w} ||\nabla h_s||^{p_w} \nabla h_s\right) > 0, \tag{23}$$

They both ensure that model (3) is well-posed by enforcing that a downflow direction exists everywhere, at the price of introducing quite stringent restrictions on the admissible topographies. Moreover, the convergence theory of [11] is established assuming such a condition on the topography. Notice that they are **sufficient conditions**, and not necessary ones: this implies that solutions to (1) and (3) can still exist for some topographies not fulfilling one of the sufficient conditions, and numerical convergence of the consistent water flux can still be observed. In particular, saddle-point or valley-like topographies will not easily fulfill those conditions, while it seems reasonable to assume that a solution will exist in such configurations since water can find a downflow direction. This being said, those probably too strong mathematical requirement should act as a warning, as it

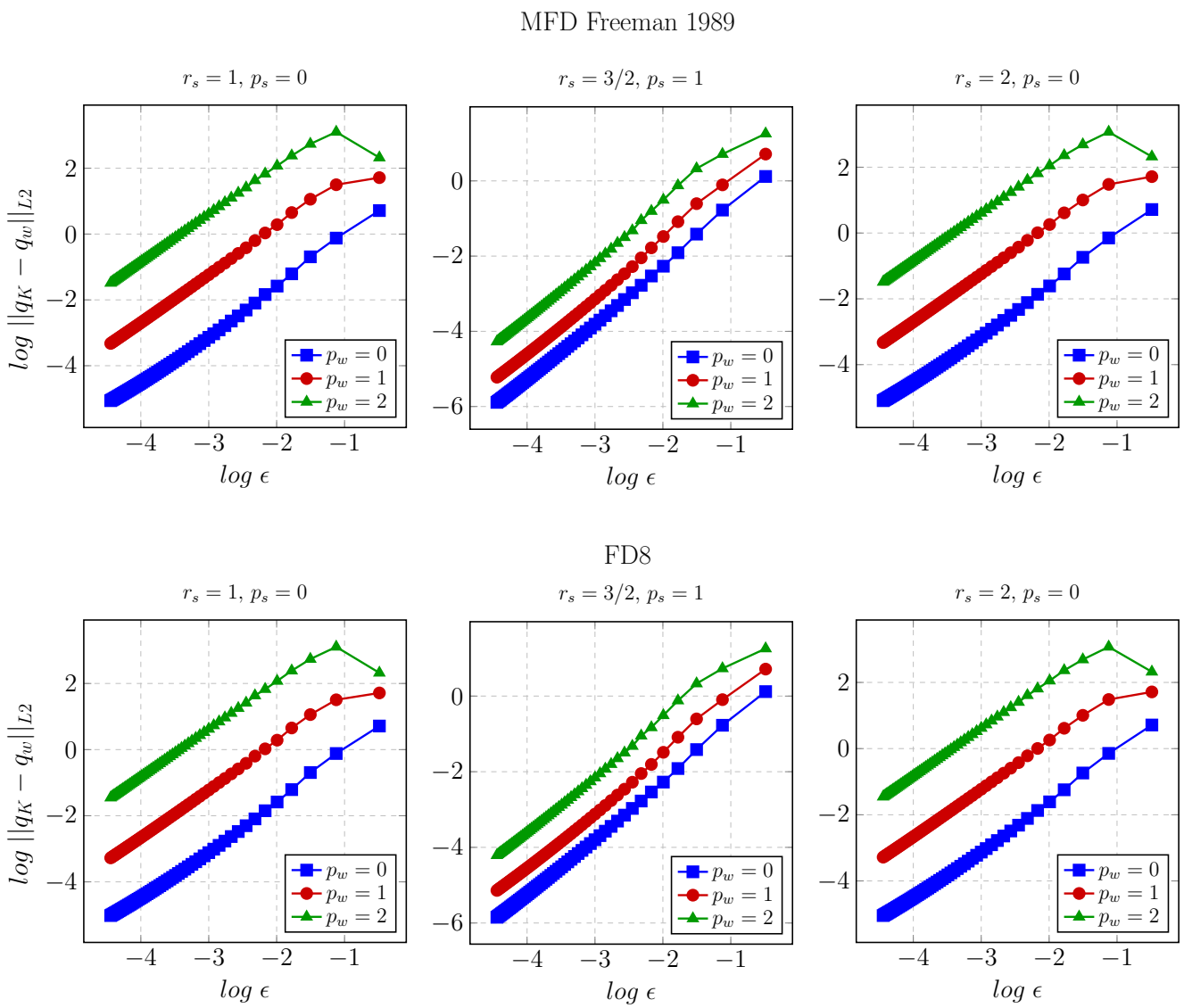



Figure 23: Convergence curves for $q_{i,j}$ towards $q_w$ for the second test configuration, for each value of $p_w$ and for each value of the couple $(r_s, p_s)$

clearly reveals that not all topographies may be admissible for model (1) and its generalization (3).

Using the notations of section 2.2, this is reflected at the discrete level by the cancellation of $s_K$ for certain configurations. Coefficient $s_K$ will cancel for a cell $K$ such that all its neighboring cells $L$ are such that $h_{s,L} \geqslant h_{s,K}$. Since $s_K = 0$, it prevents us from distributing water outside of $K$. For such a cell $K$, the MFD will lead to an abrupt stop of the water flow in cell $K$. Moreover this prevents to recompute a correct approximation of $h_{w,K}$ from the intermediate unknown $\widetilde{q}_K$ used in MFD algorithms, since we cannot invert $\widetilde{q}_K = s_K h_{w,K} \eta_w(h_{w,K})$. Such configurations correspond mostly to flat areas as well as accumulation areas. We can infer that this is a discrete indicator of what could be the weakest theoretical requirements on the topography for models (1) and

(3) to be well posed: the absence of flat or accumulation areas. One way to circumvent those limitations is to resort to more involved non-linear water flow models such as the one proposed in [12].

## Conclusion

After introducing a general Gauckler-Manning-Strickler model, we have generalized the results of [11] to a larger family of cell-to-cell MFD algorithms showing that their output coincides to an intermediate discrete quantity $\widetilde{q}_K$ occurring when applying the TPFA finite volume scheme to discretize the GMS model. Thanks to this reinterpretation, we have shown that following the idea of [11] one can reconstruct a consistent discrete water flux magnitude $q_K$ from this intermediate quantity in a post-processing step. Numerical examples illustrate that the discrete water flux magnitude is as mesh independent as one could hope for and successfully converges to the continuous water flux magnitude. This discrete water flux magnitude $q_K$ should thus be considered as the correct output of a MFD algorithm, instead of $\widetilde{q}_K$ or normalized versions of it. Then, we have extended those results to node-to-node MFD algorithms, presenting a classification of most classical algorithms of the literature using the GMS model as basis. The same post-processing correction was presented and numerical example illustrate that it again successfully solve the anomalous mesh dependency issues. Finally, we have recalled the main limitations of the GMS model, emphasizing in particular through our reinterpretation of the MFD algorithms that flat or accumulation areas require special treatments beyond the MFD algorithms. We believe that the post-processing correction presented here can benefit existing software/models relying on MFD algorithms.

## References

[1] J. J. Armitage. Short communication: flow as distributed lines within the landscape. *Earth Surface Dynamics*, 7(1):67–75, 2019.

[2] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.

[3] D. Banninger. Technical note: Water flow routing on irregular meshes. *Hydrol. Earth Syst. Sci., Vol. 11, pp. 1243-1247*, 2007.

[4] C. Bardos. Problèmes aux limites pour les équations aux dérivées partielles du premier ordre à coefficients réels ; théorèmes d'approximation ; application à l'équation de transport. *Ann. Sci. Ec. Norm. Sup. Ser. 4, Vol. 3, pp. 185-233*, 1970.

[5] P. D. Bates, M. S. Horritt, and T. J. Fewtrell. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387(1):33–45, 2010.

[6] D . B. Beasley, L. F. Huggins, and E. J. Monke. Answers: A model for watershed planning. *Transactions of the ASAE, Vol. 23(4), pp. 938-944*, 1980.

[7] B. Birnir, T. R. Smith, and G. E. Merchant. The scaling of fluvial landscapes. *Comput. Geosci.*, 27(10):1189–1216, 2001.

[8] S. Bonetti, A. D. Bragg, and A. Porporato. On the theory of drainage area for regular and non-regular points. *Proc. R. Soc. A 474: 20170693*, 2018.

[9] J. Braun and S. D. Willett. A very efficient o(n), implicit and parallel method to solve the stream power equation governing fluvial incision and landscape evolution. *Geomorphology*, 180-181:170–179, 2013.

[10] S. Carretier, L. Guerit, R. Harries, V. Regard, P. Maffre, and S. Bonnet. The distribution of sediment residence times at the foot of mountains and its implications for proxies recorded in sedimentary basins. *Earth and Planetary Science Letters*, 546:116448, 2020.

[11] J. Coatléven. Some multiple flow direction algorithms for overland flow on general meshes. *ESAIM: Mathematical Modelling and Numerical Analysis, Vol. 54 (6), pp. 1917-1949*, 2020.

[12] J. Coatléven and B. Chauveau. Large structure simulation for landscape evolution models. *Earth Surface Dynamics*, 12(5):995–1026, 2024.

[13] M. C. Costa-Cabral and S. J. Burge. Digital elevation model networks (demon): A model of flow over hillslopes for computation of contributing and dispersal areas. *Water resources research, Vol. 30(6), pp. 1681-1692*, 1994.

[14] T. J. Coulthard, J. C. Neal, P. D. Bates, J. Ramirez, G. A. M. de Almeida, and G. R. Hancock. Integrating the lisflood-fp 2d hydrodynamic model with the caesar model: implications for modelling landscape evolution. *Earth Surface Processes and Landforms*, 38(15):1897–1906, 2013.

[15] P. Davy, T. Croissant, and D. Lague. A precipiton method to calculate river hydrodynamics, with applications to flood prediction, landscape evolution models, and braiding instabilities. *Journal of Geophysical Research: Earth Surface*, 122(8):1491–1512, 2017.

[16] P. J. J. Desmet and G. Govers. Comparison of routing algorithms for digital elevation models and their implication for predicting ephemeral gullies. *Int. J. Geo. Inf. Syst., Vol. 10(3), pp. 311-331*, 1996.

[17] R.J. DiPerna and P.L. Lions. Ordinary differential equations, transport theory and sobolev spaces. *Invent. Math. 98, 511-547*, 1989.

[18] David L. Egholm, Mads F. Knudsen, Chris D. Clark, and Jerome E. Lesemann. Modeling the flow of glaciers in steep terrains: The integrated second-order shallow ice approximation (isosia). *Journal of Geophysical Research: Earth Surface*, 116(F2), 2011.

[19] R. H. Erskine, T. R. Green, J. A. Ramirez, and L. H. MacDonald. Comparison of grid-based algorithms for computing upslope contributing area. *Water Resour. Res., Vol. 42, W09416*, 2006.

[20] J. Fairfield and P. Leymarie. Drainage networks from grid digital elevation model. *Water resources research, Vol. 27(5), pp. 709-717*, 1991.

[21] E. Fernández-Cara, F. Guillén, and R.R. Ortega. Mathematical modeling and analysis of visco-elastic fluids of the oldroyd kind. *P.G. Ciarlet, J.L. Lions (Eds.), Numerical Methods for Fluids, Part 2, in: Handbook of Numerical Analysis, vol. VIII, North-Holland, Amsterdam, pp. 543–661*, 2002.

[22] T. G. Freeman. Drainage with divergent flow over a regular grid. *Proc. 8th Biennial Conf. Simulation Society of Australia, Canberra, pp. 160-165*, 1989.

[23] T. G. Freeman. Calculating catchment area with divergent flow based on a regular grid. *Computers & Geosciences Vol. 17(3), pp. 413-422*, 1991.

[24] J. C. Gallant and M. F. Hutchinson. A differential equation for specific catchment area. *Water resources research, Vol. 47, W05535*, 2011.

[25] V. Girault and L. Tartar. $l^p$ and $w^{1,p}$ regularity of the solution of a steady transport equation. *C. R. Acad. Sci. Paris, Ser. I, Vol. 348, pp. 885-890*, 2010.

[26] W. H. Graf and M. S. Altinakar. *Hydraulique fluviale: Ecoulement et phénomènes de transport dans les canaux à géométrie simple.* Traité de génie civil, vol. 16, Presses polytechniques et universitaires romandes, 2000.

[27] D. Granjeon. *3D forward modelling of the impact of sediment transport and base level cycles on continental margins and incised valleys*, chapter 16, pages 453–472. 2014.

[28] P. Holmgren. Multiple flow direction algorithms for runoff modelling in grid based elevation models: an empirical evaluation. *Hydrological processes, Vol. 8, pp. 327-334*, 1994.

[29] N. L. Lea. An aspect driven kinematic routing algorithm. In A. J. Parsons and A.D. Abrahams, editors, *Overland Flow: Hydraulics and Erosion Mechanics*, chapter 16. Chapman and Hall, New York, 1992.

[30] L. B. Leopold, M. G. Wolman, and J. P. Miller. *Fluvial Processes in Geomorphology.* W. H. Freeman, San Francisco, California, 1964.

[31] J. C. Maxwell. On hills and dales. *Philos. Mag. J. Sci., Vol. 4/40(269), pp. 421-427*, 1870.

[32] M. Nones. On the main components of landscape evolution modelling of river systems. *Acta Geophys. Vol 68, pp 459–475*, 2020.

[33] J. F. O'Callaghan and D. M. Mark. The extraction of drainage networks from digital elevation data. *Computer vision, graphics and image processing, Vol. 28, pp. 323-344*, 1984.

[34] S. Orlandini and G. Moretti. Determination of surface flow paths from gridded elevation data. *Water Resour. Res., Vol. 45, W03417*, 2009.

[35] J. D. Pelletier. Minimizing the grid-resolution dependence of flow-routing algorithms for geomorphic applications. *Geomorphology, Vol. 122 , pp. 91-98*, 2010.

[36] J.D. Pelletier. 2.3 fundamental principles and techniques of landscape evolution modeling. In John F. Shroder, editor, *Treatise on Geomorphology*, pages 29–43. Academic Press, San Diego, 2013.

[37] K. Perlin. An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 19(3):287–296, 1985.

[38] J. T. Perron, J. W. Kirchner, and W. E. Dietrich. Formation of evenly spaced ridges and valleys. *Nature, Vol. 460, pp. 502-505*, 2009.

[39] A. Porporato. Hydrology without dimensions. *Hydrol. Earth Syst. Sci., Vol. 26, pp. 355-374*, 2022.

[40] C. Qin, A.-X. Zhu, T. Pei, B. Li, C. Zhou, and L. Yang. An adaptive approach to selecting a flow-partition exponent for a multiple-flow-direction algorithm. *International Journal of Geographical Information Science Vol. 21(4), pp. 443-458*, 2007.

[41] P. Quinn, K. Beven, P. Chevallier, and O. Planchon. The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. *Hydrological processes, Vol. 5, pp. 59-79*, 1991.

[42] P. Quinn, K. Beven, and R. Lamb. The $\ln(a \tan \beta)$ index : how to calculate it and how to use it within the topmodel framework. *Hydrological processes, Vol. 9, pp. 161-182*, 1995.

[43] A. Richardson, C. N. Hill, and J. T. Perron. Ida: An implicit, parallelizable method for calculating drainage area. *Water Resour. Res. Vol. 50, pp. 4110-4130*, 2014.

[44] T. Salles. escape: parallel global-scale landscape evolution model. *Journal of Open Source Software*, 3(30):964, 2018.

[45] T. Salles, N. Flament, and D. Müller. Influence of mantle flow on the drainage of eastern australia since the jurassic period. *Geochemistry, Geophysics, Geosystems*, 18(1):280–305, 2017.

[46] J. M. Schoorl, M. P. W. Sonneveld, and A. Veldkamp. Three-dimensional landscape process modelling: the effect of dem resolution. *Earth Surface Processes and Landforms, Vol 25(9), pp 1025-1034*, 2000.

[47] J. Seibert and B. L. McGlynn. A new triangular multiple flow direction algorithm for computing upslope areas from gridded digital elevation models. *Water Resour. Res., Vol. 43, W04501*, 2007.

[48] D. G. Tarboton. A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water resources research, Vol. 33(2), pp. 309-319*, 1997.

[49] G. E. Tucker and G. R. Hancock. Modelling landscape evolution. *Earth Surface Processes and Landforms*, 35(1):28–50, 2010.

[50] D. Valters. *Modelling Geomorphic Systems: Landscape Evolution*, page 6.5.12. 05 2016.

[51] P. A. Van der Beek. *Environmental Modelling: Finding Simplicity in Complexity*, pages 309–332. 2nd Edition (eds. J. Wainwright and M. Mulligan), Wiley, Chichester, 2013.

[52] H. Beirão Da Veiga. Existence results in sobolev spaces for a stationary transport equation. *Ricerche Mat. Suppl. XXXVI pp. 173-184*, 1987.

[53] G. Willgoose. *User manual for SIBERIA (Version 8.30)*. 2005.

[54] G. Willgoose, R.L. Bras, and I. Rodriguez-Iturbe. *A physically based channel networkand catchment evolution model.* PhD thesis, TR 322, Ralph M. Parsons Laboratory, Dept. ofCivil Engineering, MIT, Boston, M, 1989.

[55] Garry Willgoose, Rafael L. Bras, and Ignacio Rodriguez-Iturbe. A coupled channel network growth and hillslope evolution model: 1. theory. *Water Resources Research*, 27(7):1671–1684, 1991.

[56] J. P. Wilson, G. Aggett, Y. Deng, and C. S. Lam. *Water in the Landscape: A Review of Contemporary Flow Routing Algorithms*, pages 213–236. Qiming Zhou and Brian Lees and Guo-an Tang. Springer Berlin Heidelberg, Springer Berlin, Heidelberg, 2008.

[57] Q. Zhou, P. Pilesjö, and Y. Chen. Estimating surface flow paths on a digital elevation model using a triangular facet network. *Water Resour. Res., Vol. 47, W07522*, 2011.