Memo

To
Ap van Dongeren, Bert Jagers

Date	**Reference**	**Number of pages**
Dec 2019		12
From	**Direct line**	**E-mail**
Mart Borsboom	+31 (0)88 335 8435	mart.borsboom@deltares nl

Subject
Design of an improved numerical implementation for the Exner equation.
Version 0.3 with adaptations based on the review by Victor Chavarrias.

Copy to
Victor Chavarrias

1 Introduction/summary

It has been noticed that the numerical implementation of the Exner equation currently applied in Delft3D (and in D-Flow FM, I presume) is intrinsically unstable. This affects the stability, the efficiency (because it is responsible for a numerical limitation to the MorFac MF), and the accuracy of morphological simulations.

A 1D Fourier-mode analysis[1] has revealed the cause. It is mentioned for the first time in Borsboom (2015) (still to be documented/published properly): the numerical implementation of the Exner equation uses a first-order accurate one-sided discretization that is upwind in the direction of the flow and the sediment transport. Because of the low accuracy, it introduces a large amount of numerical dissipation in the transport of bottom perturbations. However, the one-sided scheme introduces an *unstable* downwind component in the wave mode propagating in the opposite flow direction[2]. Because the instability scales with MF (and with Fr^4), it can be kept small by keeping MF small. The instability can be suppressed by increasing the dissipation, e.g., by means of an artificial (non-physical) enhancement of the viscosity, which Delft3D users with morphological applications seem to do frequently[3]. More information in Section 3.

[1] An analysis of the 2D case has not been performed yet and might reveal specific 2D-related aspects. On the other hand, the Fourier-mode analysis applied in 1D would in 2D be applied in an arbitrary direction and hence is in essence a 1D analysis. A 2D analysis will therefore largely lead to similar insights and conclusions.

[2] It is not an option to use a central discretization of the Exner equation instead. Because first-order accurate forward Euler in time is used (the simplest possible time integration), this would give very poor stability properties and large discretization errors (in time). Changing the time integration to, for example, leap-frog (in combination with a higher-order wiggle filter) would be an option, but is not easy to realize within the Delft3D (and D-Flow FM) framework

[3] Because viscosity in D-Flow FM is explicit (in Delft3D it is implicit) and hence cannot be set arbitrarily large because of stability restrictions, this trick is not well possible in D-Flow FM. This is at least one of the reasons why in D-Flow FM MF cannot be taken as large as in Delft3D, generally speaking.

IMPORTANT: according to the D-Flow FM Technical Reference Manual, the mass-conservative continuity equation of D-Flow FM, just like the one used in XBeach (however, this may not be true, cf. Footnote 6) but *unlike* the form used in Delft3D, *includes* the variation of the bottom. Our analyses have shown that this leads to much larger MF–dependent errors in the continuous formulation, Borsboom (2015). We have never investigated what it does to the numerical errors. It is reasonable to assume that when MF–dependent errors at the continuous level are much larger, MF–dependent

About a year ago we thought of a way to construct (by approximation, cf. the details in Section 3) a genuine upwind scheme for the Exner equation, i.e., a scheme that is stable for *all* solution modes, despite the explicit forward Euler time integration. An analysis (to be reported/published somewhere) has confirmed that the scheme is not only stable for the transport of bed perturbations, but also does not introduce unstable behavior in the flow part of the system of equations. The analysis shows that this largely removes the numerical stability limits on MF. Moreover, the new approach opens the possibility to apply higher-order upwind schemes (with flux limiter to suppress non-physical over- and undershoots). This gives very large improvements in accuracy on the relatively coarse grids that are usually applied.

In section 2 we present and analyse the standard class of flux-limited explicit upwind schemes for a scalar transport equation and write it in a form suitable for application to the Exner equation. In section 3 we briefly explain why the straightforward one-sided space discretization of the Exner equation in Delft3D (and D-Flow FM) is *not* an upwind discretization and in fact bad numerical modeling practice, and we show how a true upwind scheme for the Exner equation (with obviously much better performance) can be constructed. In section 4 the correct order of the time integration of the flow equations and the Exner equation (coupled, but performed in separate steps) is addressed.

The next step is a proof of concept of the proposed numerical implementation of the Exner equation by means of a series of tests, using the simplest possible software implementation in Delft3D. Upon success the full implementation in D-Flow FM can be considered.

NB, we have not yet looked at the application of the new scheme to the bed-load transport of graded sediment, i.e., at an improved numerical implementation of the Exner equation combined with a Hirano-type model. We expect that an approach similar to the one described in this memo will lead to similar improvements.

2 Preliminaries

Consider the 1D scalar transport equation in conservative form:

$$\frac{\partial c}{\partial t} + \frac{\partial F}{\partial x} = 0 \,, \tag{1}$$

with flux $F = u\,c$, i.e., scalar quantity c is transported with velocity u. Transformed to computational space with $x = x(\xi)$ where ξ is the space coordinate in computational space (cf. for example Mooiman (2012)), this equation becomes:

$$x_\xi \, \frac{\partial c}{\partial t} + \frac{\partial F}{\partial \xi} = 0 \,, \tag{2}$$

discretization errors (and hence MF–dependent unstable numerical behavior) will also be much larger. This would be another reason why in D-Flow FM MF cannot be taken as large as in Delft3D, generally speaking.

In D-Flow FM 1D, 2D and 3D modeling is taken care of by roughly the same code. Since in 1D the time derivative in the continuity equation must be $\partial A/\partial t$ (the numerical implementation of $W\partial\zeta/\partial t$ would lead to unacceptable mass conservation errors), the time derivative in 2D/3D must probably be modeled by $\partial h/\partial t$, also in order to be able to handle partially dry oblique bottom tiles. It should be possible to change that to (effectively) $\partial\zeta/\partial t$, in accordance with Delft3D, by calculating the old volume V^n in the time discretization $(V^{n+1} - V^n)/\Delta t$ using the new bottom geometry or by calculating the new volume V^{n+1} using the old bottom geometry. If not, then this has consequences for the modeling of morphodynamics in D-Flow FM versus the way it is modeled in Delft3D.

where x_ξ denotes $\partial x/\partial\xi$.

The explicit and stable higher-order upwind discretization of (2) reads[4]:

$$\Delta x_i \frac{c_i^{n+1}-c_i^n}{\Delta t} + F^n_{i+1/2} - F^n_{i-1/2} = 0\,, \tag{3}$$

with i the index of the grid cell (finite volume) with faces $\xi_{i-1/2}$ and $\xi_{i+1/2}$ in computational space, corresponding to the faces $x_{i-1/2}$ and $x_{i+1/2}$ in physical space. Coefficient $\Delta x_i = \int_{\xi_{i-1/2}}^{\xi_{i+1/2}} x_\xi \,\mathrm{d}\xi = x_{i+1/2} - x_{i-1/2}$ represents the size of grid cell/finite volume i in physical space. The upwind discretization of the fluxes in (3) reads:

$$F^n_{i+1/2} = u^n_{i+1/2} \begin{cases} c_i^n + \dfrac{1-\sigma^n_{i+1/2}}{2}\,\phi(r_i^n)\,(c^n_{i+1}-c^n_i) & ,\; u^n_{i+1/2} \geq 0 \\ c^n_{i+1} + \dfrac{1-\sigma^n_{i+1/2}}{2}\,\phi(1/r^n_{i+1})\,(c^n_i - c^n_{i+1}) & ,\; u^n_{i+1/2} < 0 \end{cases}\,, \tag{4}$$

where $\sigma^n_{i+1/2} = 2|u^n_{i+1/2}|\Delta t/(\Delta x_i + \Delta x_{i+1})$ is the Courant number and where ϕ is the limiter function that is a function of slope ratio $r_i^n = (c_i^n - c_{i-1}^n)/(c_{i+1}^n - c_i^n)$. Taking into account the grid variation the expression for the slope ratio reads:

$$r_i^n = \frac{(c_i^n - c_{i-1}^n)/(\Delta x_{i-1} + \Delta x_i)}{(c_{i+1}^n - c_i^n)/(\Delta x_i + \Delta x_{i+1})}\,. \tag{5}$$

On a smoothly varying grid, taking the grid variation into account in the slope ratio or not will have a minor effect. The situation is different for the (at least locally) strongly varying grids typically used in D-FLOW FM applications, where we also have to deal with non-aligned variables due to the (locally) unstructured grid and the (local) grid irregularities. It seems that taking the grid variation into account gives a more reliable slope ratio. We have not checked yet if that may conflict with the TVD properties of the scheme. Publications to be consulted here are for example Hou *et al.* (2012, 2014, 2015); Zhao *et al.* (2018, 2019). This work, apparently supervised by Hinkelmann, has been used by Sander Van Der Pijl to construct suitable flux limiters for D-FLOW FM, so this literature is something to have a look at.

The flux limiters that we will consider are:

$$\begin{aligned} &\text{Koren:} && \phi(r) = \max\big(0, \min(2r, (2+r)/3, 2)\big)\,, \\ &\text{Monotoniced Central (MC):} && \phi(r) = \max\big(0, \min(2r, (1+r)/2, 2)\big)\,, \end{aligned} \tag{6}$$

which define the flux-limited versions of the third-order upwind scheme (obtained by inserting $\phi(r) = (2+r)/3$ in (4)) and the Fromm scheme (obtained by inserting $\phi(r) = (1+r)/2$ in (4)). Other

[4] It seems to be common practice to ignore in such discretizations the spatial variation of u and x_ξ to some extent. This is mainly possibly an issue in 2D. In 2D, (the equivalent of) volume size Δx_i in one computational coordinate direction is (usually) averaged per finite volume in the other computational coordinate direction and hence approximated with (at most) second-order accuracy. Likewise in 2D, face-averaged velocities like $u^n_{i+1/2}$ are multiplied by face-averaged approximations of $c^n_{i+1/2}$ in flux discretizations like (4), ignoring the contribution of the product of the linear variation of $u^n_{i+1/2}$ and $c^n_{i+1/2}$ to the integral of flux $u\,c$ over volume faces. This limits the accuracy of the scheme to second order, regardless of the order of accuracy with which $c^n_{i+1/2}$ is approximated.

Note that when the grid and velocity field vary smoothly in computational space (physical coordinates and velocity components are smooth functions of the computational coordinates, which implies the use of a smoothly varying structured grid), these second-order errors are relatively small. In that case it makes sense to use a higher-order discretization of $c^n_{i+1/2}$, in order to obtain a high numerical accuracy in case of strongly varying c on relatively coarse grids.

substitutions give other well-known schemes, such as first-order upwind ($\phi(r) = 0$), second-order upwind ($\phi(r) = r$), and Lax-Wendroff ($\phi(r) = 1$).

A Fourier-mode analysis can be used to get some idea about the accuracy of these schemes. By inserting Fourier mode $c(x,t) = C(t)\exp(ikx)$ in (1), and by assuming constant coefficients and a uniform grid, the exact phase change $g_{\text{exact}} = \ln(c^{n+1}/c^n) = -ik\Delta x\,\sigma$ over a time step of size Δt is obtained, where σ denotes the (signed) Courant number $\sigma = u\Delta t/\Delta x$. Inserting the same Fourier mode in a numerical implementation of (1) gives $g_{\text{num}} = -ik\Delta x\, b - k\Delta x\, a$, with b an approximation of σ and a the numerical damping introduced by the scheme; b and a are a function of the type of scheme, Courant number σ, and dimensionless Fourier mode $k\Delta x$. We define the relative phase error $(b-\sigma)/\sigma$ and the relative amplitude error a/σ. They are shown for the schemes mentioned above as a function of $k\Delta x$ for two different values of σ in the Figures 1 and 2.

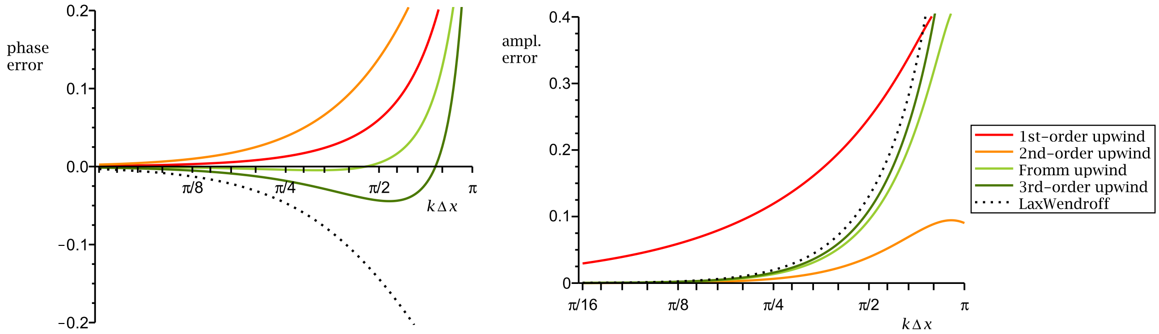

Figure 1: Result of Fourier-mode analysis of explicit scheme (3) at high Courant number $\sigma = 0.7$ for a number of flux approximations (4). Left relative phase error, right relative amplitude error. Figures taken from Borsboom (2016, 2018, 2019).

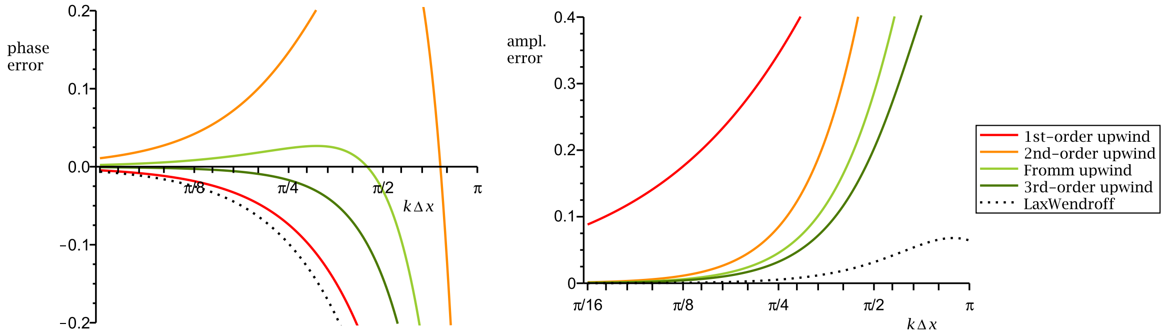

Figure 2: Result of Fourier-mode analysis of explicit scheme (3) at low Courant number $\sigma = 0.1$ for a number of flux approximations (4). Left relative phase error, right relative amplitude error. Figures taken from Borsboom (2016, 2018, 2019).

For $\sigma^n_{i+1/2} = 1$, flux (4) becomes independent of the flux limiter. For constant u and Δx, and $\sigma = 1$, flux (4) reduces to an expression that can be written as $F^n_{i+1/2} = (u/2 + |u|/2)\,c^n_i + (u/2 - |u|/2)\,c^n_{i+1}$, and scheme (3) reduces to $c^{n+1}_i = c^n_{i-1}, u > 0$ and $c^{n+1}_i = c^n_{i+1}, u < 0$, i.e., 'exact' cell-to-cell transfer of the c^n_i to the values c^{n+1}_i at the next time level when $\sigma = 1$. This is why the schemes are relatively accurate at values of σ that are close to 1, such as in Figure 1. However, since in practice u and Δx typically show large variations while σ cannot be larger than 1 because of stability restrictions, σ in applications is typically much smaller than 1. Figure 2 therefore gives a better impression of the error behavior of the schemes encountered in practice.

Because of the large amplitude error of the dissipative first-order upwind scheme at $\sigma = 0.1$ (red line in right Figure 2), this scheme is only 5% accurate for $k\Delta x \lessapprox \pi/28$, i.e., perturbations need to be resolved with at least about $2 \times 28 = 56$ grid cells to reach this level of accuracy. The Fromm scheme has at $\sigma = 0.1$ an accuracy of 5% or better for all Fourier modes $k\Delta x \lessapprox \pi/4$, while for the third-order upwind scheme this level of accuracy is reached for all $k\Delta x \lessapprox \pi/3$. In other words, to ensure errors of less than 5%, perturbations need to be resolved with at least $2 \times 4 = 8$ grid cells (Fromm scheme) or $2 \times 3 = 6$ grid cells (third-order upwind scheme). These numbers have been obtained by considering the relative total discretization error (phase error and amplitude error combined), defined as $|ib + a - i\sigma|/\sigma$.

The results presented in the previous paragraph clearly show the advantage of using higher(-order) accurate schemes: the same accuracy on a much coarser grid. They indicate that compared to first-order upwind, $56/8 = 7$ (Fromm) to $56/6 = 9$ (third-order upwind) times less grid points (per direction!) would be required. In practice the gain in accuracy will be smaller due to things like the variation in grid size/grid structure, the non-uniform velocity, and the numerical implementation at boundaries (higher-order methods are more sensitive to these things than lower-order methods).

Another reason why the gain in accuracy will be smaller in practice is the flux limiter. The purpose of a flux limiter is to ensure TVD (Total Variation Diminishing) behavior that keeps the solution monotone and free of spurious over- and undershoots. Unfortunately, it also cuts off perfectly smooth local maxima and minima, as mentioned in the Introduction of Shu (1997) for example. Because the slope of the solution changes sign in extrema, the slope ratio (5) is always negative at a maximum or a minimum, setting flux limiters like those in (6) locally to zero and hence flux (4) locally to first-order upwind. The result is a spurious flattening of smooth maxima and minima.

When a scheme reduces to first-order upwind at the one or two grid cells at/around a smooth local minimum or maximum, the global order of accuracy of that scheme must be limited to two, regardless of the order of accuracy of the underlying (very) high accurate upwind scheme[5]. In fact, all measures that ensure a certain degree of smoothness, boundedness and/or positivity do so by adding some low-pass filter mechanism (flux limiters, WENO, higher-order artificial diffusion, . . .) to a higher-order accurate discretization. This will always be at the expense of some accuracy.

In (4) the flux is written as the sum of first-order upwind plus a (flux-limited) higher-order correction. For use in the next section we write it as the sum of a central part and a (flux-limited) (higher-order)

[5] Flux limiters ensure oscillation-free solutions that are pleasing to the eye. They do not necessarily ensure high accuracy. Techniques exist that break through the order-of-accuracy barrier of flux-limited higher-order accurate upwind schemes, for example WENO, cf. Shu (1997), and possibly also DG (I am not familiar with those methods).

NB, I have checked the paper Qiu and Shu (2011) where third- and fifth-order accuracy is reported for (W)ENO3 and WENO5 applied to 2D linear transport. Because ENO and WENO are not TVD but 'only' (Weighted) Essentially Non-Oscillatory, they allow higher than second-order accuracy. NB2, in the reported 2D linear transport test a *constant* velocity field and a *uniform* grid are used, i.e., that test says *nothing* about the order of accuracy of the scheme on a non-uniform grid and for a variable velocity field.

dissipative part:

$$F_{i+1/2} = u_{i+1/2}\,\frac{c_i + c_{i+1}}{2} + u_{i+1/2}\begin{cases} \dfrac{(1-\sigma_{i+1/2})\,\phi(r_i) - 1}{2}\,(c_{i+1} - c_i) & ,\ u_{i+1/2} \geq 0 \\ \dfrac{(1-\sigma_{i+1/2})\,\phi(1/r_{i+1}) - 1}{2}\,(c_i - c_{i+1}) & ,\ u_{i+1/2} < 0 \end{cases} . \tag{7}$$

3 A stable numerical implementation of the Exner equation

Consider the 1D Exner equation:

$$(1-\varepsilon)\frac{\partial z_b}{\partial t} + MF\frac{\partial S}{\partial x} = 0 \, . \tag{8}$$

This equation is of the form (1), but *not* with a flux that can be written as $F = u\,c$! Sediment transport S is (mainly) a function of flow velocity u, for example Engelund-Hansen:

$$S = \alpha_B S_{EH} \ , \ \text{with} \ \ S_{EH} = \frac{0.05\,\alpha u^5}{\sqrt{g}\,C^3 \Delta^2 D_{50}} \ , \tag{9}$$

and with coefficient α_B the Bagnold correction for the bed-slope effect:

$$\alpha_B = 1 + \alpha_{BS}\left[\frac{\tan\phi}{\cos(\arctan(\partial z_b/\partial x))(\tan\phi + \partial z_b/\partial x)} - 1\right] . \tag{10}$$

Observation: (8) with S_{EH} as in (9) is *not* a transport equation for scalar quantity z_b (with a diffusion part because of the bed-slope-dependent Bagnold correction), but a model equation for the bed evolution that is strongly coupled to the 1D shallow-water flow equations.

To save time I have left out in the two paragraphs below the detailed expressions for the moment. This is acceptable, since they are not relevant for the purpose of this memo. The required expressions are somewhat hidden in my Maple worksheets with Exner and MorFac analyses, and have not yet all been extracted for the purpose of our MorFac paper in the making. It takes time to do so, which I decided to skip for the moment.

An analysis of the full system of coupled DELFT3D model equations (continuity equation, momentum equation, and Exner equation) for *subcritical flow* (Fr sufficiently smaller than 1) shows that the system is hyperbolic (when dissipation terms are neglected) up to some MF that is too large to be practically useful[6] (hence not a disadvantage), cf. Borsboom (2015) and our MorFac paper that is still in preparation. The eigenvalue-eigenvector structure of the system matrix shows that, as

[6] This is due to the form of the DELFT3D continuity equation that is not strictly conservative. The XBEACH system of model equations, with a continuity equation in mass conservative form, is hyperbolic and well posed for any value of MF (as far as we have verified, i.e., at least for low subcritical flow speeds), cf. Borsboom (2015) and our MorFac paper that is still in preparation. IMPORTANT: according to the Technical Reference Manual, the continuity equation in D-FLOW FM, *unlike* the form used in DELFT3D, is *also* in mass conservative form. Our analyses have shown that this leads to much larger MF–dependent errors. Checking the XBEACH Technical Reference Manual, Roelvink *et al.* (2015), we notice that, although the continuity equation is in fully conservative form at the continuous level, cf. equation (B.4), it is *not* at the discrete level. In discretization (B.5) the time variation of the bed level is missing (just like in DELFT3D).

expected, the information transported with the bed celerity (approximately equal to $\partial S/\partial u$, but not quite) is a *combination* of z_b, u and ζ. Because of the couplings flow ⇒ sediment transport ⇒ bottom changes ⇒ flow, there is *no* separate bed-level transport equation. More importantly, the information transported with the current-wave speeds (approximately equal to the well-known $u + \sqrt{gh}$ and $u - \sqrt{gh}$) is not only a combination of flow quantities (approximately the well-known $h\,u + \sqrt{gh}\,\zeta$ and $h\,u - \sqrt{gh}\,\zeta$) but also contains a small z_b–part that scales with MF and also with Fr^4, if I remember correctly[7].

A one-sided space discretization of (8) is therefore *not* an upwind discretization. Because of the z_b–part in the eigenvector pertaining to the eigenvalue $u - \sqrt{gh}$, it introduces partial *downwind* in the characteristic propagating in opposite u–direction (when $Fr < 1$). A one-sided quasi-upwind space discretization of (8) is therefore intrinsically unstable, and in practice only stable because of the stabilizing effect of physical dissipation mechanisms in the model (bottom friction, viscosity) or the presence of numerical dissipation (the genuine upwind and/or the fully implicit time integration applied in the discretization of certain terms). This problem has been noticed before, see for example Cordier *et al.* (2011) where it is concluded that "*in order to obtain a robust numerical scheme the full system must be considered and a coupled numerical scheme must be used*". This incorrect conclusion is based on their use of a one-sided discretization when solving the Exner equation separate from the flow equations. It seems that the authors have failed to see that the unstable behavior of their split scheme is at least partially caused by the properties of their numerical implementation. See also Audusse *et al.* (2018) who "*emphasize that, in fact, they* (MB: i.e., authors who 'conclude to the bad behavior of the splitting approach') *only conclude to the bad behavior of one possible way to perform the splitting approach, not of the splitting approach itself*". This is clearly a subject that needs to be continued.

The larger Fr and MF, the larger the z_b–part in the $u - \sqrt{gh}$ characteristic (cf. Footnote 7), and the larger the destabilizing effect of a one-sided discretization of the Exner equation. To compensate for this effect, DELFT3D users tend to increase viscosity coefficients or are forced to lower the value of MF. Note that when a first-order one-sided discretization is used for the Exner equation (in DELFT3D the default option), the destabilizing effect in the negative $u - \sqrt{gh}$ direction is about maximal, but also the stabilizing effect in the positive bed-celerity direction is maximal. The latter causes the excessive spurious smoothing and smearing of (usually underresolved) bottom features like trenches and bars, cf. the text below Figure 2 where it is mentioned that perturbations that are numerically propagated by a first-order upwind scheme may have to be modeled with a resolution of at least 56 grid cells to reach an error level of 5% or less.

There would be no problem to use an upwind discretization technique for the Exner equation[8] if it could be reformulated as a separate transport equation for z_b. Because of the already mentioned coupling with the flow equations this is impossible, however, it turns out to be possible by approximation when we restrict ourselves to $Fr \ll 1$ (other Fr-ranges not considered yet), which for most

[7] This scaling comes from $\partial S_{EH}/\partial u$: S scales with Fr^5 in the Engelund-Hansen sediment transport formula (9), hence its derivative with respect to u (a coefficient in the linearization) scales with Fr^4.

NB, because of the scaling with MF and Fr^4, the z_b–part in the flow characteristics is not that small anymore at very large MF and small Fr, at large MF and moderate Fr, or at $MF = 1$ and large Fr.

[8] The proper way to apply upwind is to decompose the fluxes of the *coupled* system of flow and Exner equations in parts associated with each of the eigenvalues (flux splitting), and to apply separate upwind discretizations to the flux parts taking into account the sign of the associated eigenvalues, i.e., the well-known Riemann-solver approach. Although feasible, see, e.g., Castro Díaz *et al.* (2009); Li and Duffy (2011); Serrano-Pacheco *et al.* (2012), this couples in a very complex way the discretization of flow equations and Exner equation, which is highly undesirable.

practical applications is sufficient. The approximation is obtained by linearizing S about some state:

$$S = S_0 + \left.\frac{\partial S}{\partial u}\right|_0 (u - u_0) + \left.\frac{\partial S}{\partial h}\right|_0 (h - h_0) + \ldots \; .$$

With S as in (9) we get (linearization of other sediment transport formulations is similar):

$$S_{EH} = \frac{0.05\,\alpha u_0^5}{\sqrt{g}\,C^3 \Delta^2 D_{50}} + \frac{0.25\,\alpha u_0^4}{\sqrt{g}\,C^3 \Delta^2 D_{50}} (u - u_0) \;, \tag{11}$$

Given the fact that S_{EH} is *not* of the form of some (bottom) celerity u_b times z_b (i.e., of the form of a z_b advection term), the idea is to make S_{EH} of such a form *locally*, by writing the *variation* of S_{EH} (approximately) as a function of z_b. Bagnold correction α_B is already a function of z_b only, cf. (10), hence this coefficient remains unchanged.

Assuming steady state and neglecting the effect of the dissipation terms (bottom friction and viscosity) locally, we obtain from the continuity equation and momentum equation[9] (note that the second equation expresses the conservation of energy):

$$\begin{aligned} &0 \approx hu - h_0 u_0 \approx h_0 (u - u_0) + u_0 (\zeta - z_b - \zeta_0 + z_{b,0}) \;, \\ &0 \approx u^2/2 + g\zeta - u_0^2/2 - g\zeta_0 \approx u_0 (u - u_0) + g(\zeta - \zeta_0) \;, \end{aligned}$$

and hence:

$$u - u_0 \approx \frac{g u_0}{g h_0 - u_0^2} (z_b - z_{b,0}) \;,\; h - h_0 \approx \frac{-g h_0}{g h_0 - u_0^2} (z_b - z_{b,0}) \;. \tag{12}$$

Because we restrict ourselves to the case $Fr \ll 1$ for the moment, we have $gh_0 - u_0^2 \approx gh_0$, i.e., the divisions in these expressions are well defined.

Inserting (12) in (11) (i.e., only a term with a $u - u_0$ to be replaced), we get:

$$S_{EH} \approx S_{EH,0} + \left.\frac{\partial S_{EH}}{\partial u}\right|_0 \frac{g u_0}{g h_0 - u_0^2} (z_b - z_{b,0}) \;.$$

Inserting this expression in (8) (adding the Bagnold correction α_B), it is clear that an upwind discretization does *not* exist for the first term in the right-hand side, because this general sediment transport expression is *not* merely a function of z_b. However, the second term is formulated in z_b. An (approximate) genuine upwind discretization is therefore possible by following the recipe depicted in (7) (where we replace $c_{i+1} - c_i$ by $(z_{b,i+1} - z_{b,0}) - (z_{b,i} - z_{b,0}) = z_{b,i+1} - z_{b,i}$, etc.):

$$\begin{aligned} S_{\mathrm{upw},i+1/2} &= S_{i+1/2} \\ &+ u_{b,i+1/2} \begin{cases} \dfrac{(1 - \sigma_{b,i+1/2})\,\phi(r_{b,i}) - 1}{2} (z_{b,i+1} - z_{b,i}) & ,\; u_{b,i+1/2} \geq 0 \\ \dfrac{(1 - \sigma_{b,i+1/2})\,\phi(1/r_{b,i+1}) - 1}{2} (z_{b,i} - z_{b,i+1}) & ,\; u_{b,i+1/2} < 0 \end{cases} . \end{aligned} \tag{13}$$

[9] This works also in 2D with extra velocity variable v and extra v momentum equation. We then apply exactly the same 1D procedure, but in the direction of the bottom sediment transport, usually equal to the flow direction, i.e., the first equation becomes $0 \approx h\sqrt{u^2 + v^2} - h_0\sqrt{u_0^2 + v_0^2}$, the second equation becomes $0 \approx (u^2 + v^2)/2 + g\zeta - (u_0^2 + v_0^2)/2 - g\zeta_0$, while the additional equation reads $0 \approx v/u - v_0/u_0$. To be continued.

with bottom celerity (not scaled with $MF/(1-\varepsilon)$):

$$u_{b,i+1/2} = \left.\frac{\partial S_{EH}}{\partial u}\right|_{i+1/2} \frac{g u_{i+1/2}}{g h_{i+1/2} - u_{i+1/2}^2} \; ,$$

with bottom Courant number:

$$\sigma_{b,i+1/2} = \frac{MF u_{b,i+1/2} \Delta t}{(1-\varepsilon)\Delta x_{i+1/2}} \; ,$$

with slope ratio:

$$r_{b,i} = \frac{(z_{b,i} - z_{b,i-1})/(\Delta x_{i-1} + \Delta x_i)}{(z_{b,i+1} - z_{b,i})/(\Delta x_i + \Delta x_{i+1})} \; ,$$

and with limiter function ϕ equal to whatever you would like to use, e.g., Koren or MC, cf. (6).

4 Time integration of combined flow and Exner equations in Delft3D

In this section the time integration of the coupled shallow-water flow equations and Exner equation is given for Delft3D and not for D-Flow FM. Because so far space discretizations of the coupled shallow-water–Exner equations have only been analyzed in combination with the Delft3D time integration, it is more convenient to implement and validate the proposed improved numerical implementation for the Exner equation in Delft3D for a quick proof of concept.

The global form of the two-stage time integration of the 2DH shallow-water equations applied in Delft3D reads (cf. Mooiman (2012); all terms that are not (in)directly affected by bottom updates are omitted):

Stage 1: η–momentum equation, followed by ξ–derivative in continuity equation coupled implicitly to water-level gradient in ξ–momentum equation, solved iteratively:

$$\frac{v^{n+1/2} - v^n}{\Delta t/2} + \ldots + g \frac{\partial \zeta^n}{\partial \eta} = -g \frac{|\mathbf{u}^n| v^{n+1/2}}{C^2 h^n} \; , \tag{14}$$

$$\frac{\zeta^{n+\frac{1}{2},p} - \zeta^n}{\Delta t/2} + \frac{\partial h^{n+\frac{1}{2},p} u^{n+\frac{1}{2},p}}{\partial \xi} + \frac{\partial h^n v^n}{\partial \eta} = 0 \; , \tag{15}$$

$$\frac{u^{n+\frac{1}{2},p} - u^n}{\Delta t/2} + \ldots + g \frac{h^{n+\frac{1}{2},p-1}}{h^{n+\frac{1}{2},p}} \frac{\partial \zeta^{n+\frac{1}{2},p}}{\partial \xi} = -g \frac{|\mathbf{u}^n| u^{n+\frac{1}{2},p}}{C^2 h^n} \; , \; p = 1, 2, \ldots \; . \tag{16}$$

Stage 2: ξ–momentum equation, followed by η–derivative in continuity equation coupled implicitly to water-level gradient in η–momentum equation, solved iteratively:

$$\frac{u^{n+1} - u^{n+1/2}}{\Delta t/2} + \ldots + g \frac{\partial \zeta^{n+1/2}}{\partial \xi} = -g \frac{|\mathbf{u}^{n+1/2}| u^{n+1}}{C^2 h^{n+\frac{1}{2}}} \; , \tag{17}$$

$$\frac{\zeta^{n+1,p} - \zeta^{n+1/2}}{\Delta t/2} + \frac{\partial h^{n+1/2} u^{n+1/2}}{\partial \xi} + \frac{\partial h^{n+1,p} v^{n+1,p}}{\partial \eta} = 0 \; , \tag{18}$$

$$\frac{v^{n+1,p}-v^{n+1/2}}{\Delta t/2}+\ldots+g\frac{h^{n+1,p-1}}{h^{n+1,p}}\frac{\partial\zeta^{n+1,p}}{\partial\eta}=-g\frac{|\mathbf{u}^{n+1/2}|u^{n+1,p}}{C^2h^{n+1/2}}\ ,\ p=1,2,\ldots\ . \tag{19}$$

The water depth $h=\zeta-z_b$ in the right-hand sides of (14), (16), (17) and (19) is at the previous time levels t^n and $t^{n+1/2}$, so no problem here. Likewise for h in the third term (second term) of the right-hand side of (15) ((18)). However, in the second respectively third term of (15) and (18), a water depth at the *next* time levels $t^{n+1/2}$ and t^{n+1} is used. While water level ζ at these next time levels is determined iteratively, bottom level z_b should be available. An analysis (to be reported/published somewhere) has shown that using here z_b at the *previous* time level (as is currently the case in Delft3D where Exner equation (8) is solved *after* the flow equations) has a significant negative effect on the numerical accuracy. The discretization in time of (8) should therefore be:

$$(1-\varepsilon)\frac{z_b^{n+1/2}-z_b^n}{\Delta t/2}+MF\frac{\partial S^n}{\partial\xi}+MF\frac{\partial S^n}{\partial\eta}=0\ , \tag{20}$$

to be applied *prior* to (14), (15) and (16), and:

$$(1-\varepsilon)\frac{z_b^{n+1}-z_b^{n+1/2}}{\Delta t/2}+MF\frac{\partial S^{n+1/2}}{\partial\xi}+MF\frac{\partial S^{n+1/2}}{\partial\eta}=0\ , \tag{21}$$

to be applied *prior* to (17), (18) and (19).

In Delft3D the Exner equation is applied per stage *after* the flow equations. This order can be maintained *if* the Exner discretization applied after the *first* flow stage is actually the one to be applied *before* the second flow stage, and if after the *second* flow stage the one to be applied *before* the next first flow stage is applied. In other words, apply (21) *after* (14), (15) and (16), and apply

$$(1-\varepsilon)\frac{z_b^{n+3/2}-z_b^{n+1}}{\Delta t/2}+MF\frac{\partial S^{n+1}}{\partial\xi}+MF\frac{\partial S^{n+1}}{\partial\eta}=0 \tag{22}$$

(which is (20) shifted one level forward in time) *after* (17), (18) and (19).

Note that in this way *no* discretization of the Exner equation is applied *before* the first flow stage of the very first time step. To be consistent, the result of the Exner discretization applied *after* the first flow stage (representing the one before the second flow stage) should therefore in the first time step be discarded. Another thing to pay attention to is that the result of the Exner discretization after the *first* flow stage is now the position of the bottom at the next time level t^{n+1}. This solution, and *not* the result of the Exner discretization after the *second* flow stage, is the bed level per time step.

5 Discretization in time of the Bagnold correction

Bagnold correction (10) of sediment transport formula (9) introduces a diffusion effect in Exner equation (8). It is easy to see that that diffusion effect becomes infinitely large[10] when downward slope $\partial z_b/\partial x$ gets close to $-\tan\phi$. Because of the applied *explicit* forward Euler time integration, the numerical implementation of the Exner equation in Delft3D (and in D-Flow FM) is unable to handle

[10] I suppose that this is to model avalanching: the larger the downward slope, the faster its steepness is reduced by the Bagnold correction. As a result, it is impossible for downward slope $\partial z_b/\partial x$ to reach the value $-\tan\phi$ in the *continuous* model. This may not be, and in fact is not, true for its numerical implementation.

this situation. We assume that it is for this reason that the value of $\partial z_b/\partial x$ is limited to $-0.9\tan\phi$ in DELFT3D, cf. section 11.4.7, 'Adjustment of bedload transport for bed-slope effects' in Delft3D-FLOW (2019). There is no reason to assume that this limiter is sufficient to ensure stability, but in practice it probably is. A rather nasty property of the limiter is that it abruptly switches off the diffusion effect in the Exner equation when $\partial z_b/\partial x$ reaches the value $-0.9\tan\phi$.

The stability limit of forward Euler applied to a diffusion equation is $d\Delta t/\Delta x^2 \leq 1/2$, with d the diffusion coefficient. In the Exner equation, the diffusion coefficient scales with S_{EH} as well as with MF and increases rapidly when $\partial z_b/\partial x$ gets close to $-\tan\phi$ ($-0.9\tan\phi$). The stability condition also becomes more and more severe in case of (locally) smaller and smaller grid size. Has it ever been verified that the stability condition is always met in practice? If not, then spuriously growing bottom perturbations will occur. When this growth is bounded by the coupling of the Exner equation with the flow equations, such a phenomenon may go unnoticed. Notice that stability is necessary but not sufficient for accuracy. $d\Delta t/\Delta x^2 \ll 1/2$ may be required for accuracy, especially in the case of bottom features resolved by only a few grid cells (a situation often encountered in practice).

An easy way to solve the problems mentioned above is by integrating the Bagnold correction *implicitly* in time. In such an approach, the currently applied explicit $S^n = \alpha_B^n S_{EH}^n$ in (20) would be replaced by $S^n = \alpha_B^{n+\theta/2} S_{EH}^n$, with $1/2 < \theta \leq 1$ (similar for $S^{n+1/2}$; NB, the division by 2 in the superscript $n+\theta/2$ is to compensate for the half time step per time integration stage). This leads to a nonlinear system of equations per half time step for the unknowns $z_b^{n+1/2}$ (z_b^{n+1}) that is probably highly diagonally dominant almost everywhere. A simple iterative point Jacobi method would then be sufficient to solve the system of equations. Since the first iteration would/could correspond with the currently applied forward Euler method, a few iterations are expected to be sufficient to obtain at least a major improvement.

Similar remarks on how to deal numerically with the Bagnold correction can be found in Borsboom (2017).

6 References

Audusse, E., C. Chalons and P. Ung, 2018. "A simple three-wave approximate Riemann solver for the Saint-Venant-Exner equations." *Int. J. Numer. Methods Fluids* 87 (10): 508–528. DOI: 10.1002/fld.4500.

Borsboom, M., 2015. "MorFac nieuws, feb 2015." Notitie, februari 2015.

Borsboom, M., 2016, 2018, 2019. "Fourier-mode analysis of some explicit one-step advection schemes." MAPLE worksheet, April–June 2016, August 2018, September 2019.

Borsboom, M., 2017. "MorFac nieuws, feb 2017." Notitie, februari 2017.

Castro Díaz, M. J., E. D. Fernández-Nieto, A. M. Ferreiro and C. Páres, 2009. "Two-dimensional sediment transport models in shallow water equations. A second order finite volume approach on unstructured meshes." *Comput. Meth. Appl. Mech. Eng.* 198 (33–36): 2520–2538. DOI: 10.1016/j.cma.2009.03.001.

Cordier, S., M. H. Le and T. Morales de Luna, 2011. "Bedload transport in shallow water models:

Why splitting (may) fail, how hyperbolicity (can) help." *Adv. Water Resour.* 34 (8): 980–989. DOI: 10.1016/j.advwatres.2011.05.002.

Delft3D-FLOW, 2019. "User Manual."

Hou, J., Q. Liang, H. Zhang and R. Hinkelmann, 2014. "Multislope MUSCL method applied to solve shallow water equations." *Comput. Math. Appl.* 68 (12, Part A): 2012–2027. DOI: 10.1016/j.camwa.2014.09.018.

Hou, J., Q. Liang, H. Zhang and R. Hinkelmann, 2015. "An efficient unstructured MUSCL scheme for solving the 2D shallow water equations." *Environ. Modell. Softw.* 66: 131–152. DOI: 10.1016/j.envsoft.2014.12.007.

Hou, J., F. Simons and R. Hinkelmann, 2012. "Improved total variation diminishing schemes for advection simulation on arbitrary grids." *Int. J. Numer. Methods Fluids* 70 (3): 359–382. DOI: 10.1002/fld.2700.

Li, S. and C. J. Duffy, 2011. "Fully coupled approach to modeling shallow water flow, sediment transport, and bed evolution in rivers." *Water Resour. Res.* 47 (3): W03508. DOI: 10.1029/2010WR009751.

Mooiman, J., 2012. "Delft3D-FLOW description of numerical implementation." Technical Manual.

Qiu, J.-M. and C.-W. Shu, 2011. "Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow." *J. Comput. Phys.* 230 (4): 863–889. DOI: 10.1016/j.jcp.2010.04.037.

Roelvink, D., A. Van Dongeren, R. McCall, B. Hoonhout, A. Van Rooijen, P. Van Geer, L. De Vet, K. Nederhoff and E. Quataert, 2015. *XBeach technical reference: Kingsday Release – Model description and reference guide to functionalities.* Tech. rep., Deltares, UNESCO-IHE.

Serrano-Pacheco, A., J. Murillo and P. García-Navarro, 2012. "Finite volumes for 2D shallow-water flow with bed-load transport on unstructured grids." *J. Hydraul. Res.* 50 (2): 154–163. DOI: 10.1080/00221686.2012.669142.

Shu, C.-W., 1997. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws.* Tech. Rep. CR-97-206253, NASA.

Zhao, J., I. Özgen, D. Liang and R. Hinkelmann, 2018. "Improved multislope MUSCL reconstruction on unstructured grids for shallow water equations." *Int. J. Numer. Methods Fluids* 87 (8): 401–436. DOI: 10.1002/fld.4499.

Zhao, J., I. Özgen-Xian, D. Liang, T. Wang and R. Hinkelmann, 2019. "An improved multislope MUSCL scheme for solving shallow water equations on unstructured grids." *Comput. Math. Appl.* 77 (2): 576–596. DOI: 10.1016/j.camwa.2018.09.059.