Boosted robustness of semi-implicit subgrid methods for shallow water flash floods in hills

Stelling, Guus S.

DOI
10.1016/j.compfluid.2022.105645

Publication date
2022

Document Version
Final published version

Published in
Computers and Fluids

Computers and Fluids 247 (2022) 105645

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Boosted robustness of semi-implicit subgrid methods for shallow water flash floods in hills

Guus S. Stelling

TU Delft, Stevinweg 1, 2628 CN Delft, The Netherlands

ARTICLE INFO

Keywords:
Accuracy
Coarse grid
Digital elevation models, Semi-implicit time integration
Numerical stability
Overland flow
Predictor-corrector
Rainfall-runoff
Shallow water equations
Subgrid
Thin layer calculation
Wetting and drying

ABSTRACT

The widespread availability of high-resolution Digital Elevation Models (DEM), has led to the development of subgrid numerical modeling techniques, based on Shallow Water Equations (SWE). Detailed DEM data is clustered as much as possible within a coarse grid cell that is preferably much larger than a raster pixel. This has considerable advantages for model efficiency, in particular for flood mapping. But overland flow on hills, key to rainfall-runoff, may have several problems with accuracy and stability. These issues arise especially during downhill flooding and with surface runoff on inclined planes. It is the focus of this paper. As robust solutions we propose: (1) a special *volume correction equation* with intrinsic wetting and drying but without stability constraints and, (2) a simple *thin layer calculation* that is accurate for runoff on coarse grids with sloping subgrids. Especially the combination (1) and (2) makes the subgrid method highly efficient on slopes, as is demonstrated by a few examples.

1. Introduction

Floods are considered as one of the most common and most damaging disasters caused by nature. Accurate flood predictions are crucial for the design and operation of flood protection systems [27]. Floods have a variety of causes, such as storm surge, dam failure, tsunamis, extreme rainstorms, etc. They are often due to a mix of factors. This paper focuses on the efficiency of rain flood simulations in hilly terrain.

The accessibility of high-resolution digital elevation models has triggered the development of sub-grid methods. We confine ourselves to semi-implicit subgrid methods with staggered grids [4,5,6,11,18]. Currently, there are numerous applications for different flood types [1,17,24]. However, like any method, also this method has pitfalls that may restrict the accuracy, in particular for relative coarse grids. One example is that flow blocking obstacles, like thin floodwalls, may be overlooked [11]. A number of methods are published to repair all sorts of unwanted subgrid effects [12,22]. In particular *grid cloning* deserves attention [4,7]. All these methods are essentially compensations for lack of resolution. This is a classical modeling problem, like the closure problem of turbulence. In [12] subgrids are studied from that perspective. *There is no general solution for lack of resolution*, but in each case there may be a special remedy. From this point of view, subgrid methods are no exception.

This paper is about remedies for flash flood computations in hilly areas. These areas have slopes with storm water flowing downward, while inundating lower laying areas. On shallow slopes, the elegant wetting and drying procedure from [5], sometimes requires extremely small time steps for stability. This is due to semi-implicit time integration of super critical thin layer flow, with explicit wet cross sections. It will be argued that flooding on slopes, with detailed subgrids, is a stiff problem, requiring implicit time integration [14]. We propose a non-iterative and non-conservative semi-implicit predictor, followed by a mass-conservative corrector without time step restrictions. This corrector will guarantee positive volumes, in such a way that special wetting and drying procedures are not required. Another problem of concern is the accuracy of overland flow on hills, where subgrids may have an adverse impact. For real-life applications however, other elements, such as ponding, channel flow, etc., are equally important. Here subgrids have great advantages. Thus, we propose a method to keep these advantages, but without accuracy loss on slopes.

This paper has the following sections: Section 2 is a summary of the semi-implicit subgrid method. Section 3 shows the time step limitation on slopes with thin layers of water, based on a simple example. This

Abbreviations: SWE, Shallow Water Equations.
E-mail address: gustaaf_stelling@hotmal.com.

https://doi.org/10.1016/j.compfluid.2022.105645
Received 15 February 2022; Received in revised form 8 June 2022; Accepted 24 August 2022
Available online 27 August 2022

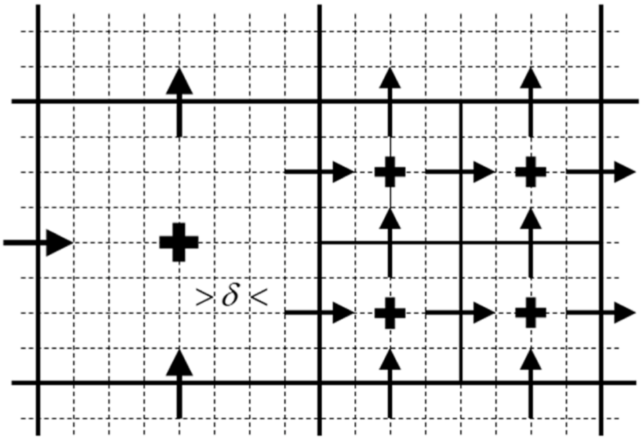

Fig. 1. Staggered quadtree grid with subgrid.

example is also used to demonstrate accuracy issues. Section 4 outlines a predictor-corrector method, with a stable, positive and conservative corrector named *Volume Correction Equation* or VCE. Section 5 explains *Thin Layer Calculation* or TLC, for accurate overland flow on hills. Section 6 describes numerical examples. Section 7 contains conclusions. The appendix calculates the time step limitation of the semi-implicit method, based on Von Neumann stability analysis of the frozen coefficient equation.

2. Summary of the subgrid method

SWE, as applied in this paper, are given by:

$$\text{momentum eqs:}\ \begin{cases} u_t + uu_x + vu_y + g\zeta_x + \dfrac{c_f}{h}u\|\ u\ \| = 0 \\ v_t + uv_x + vv_y + g\zeta_y + \dfrac{c_f}{h}v\|\ u\ \| = 0 \end{cases} \tag{1}$$

$$\text{continuity eq:}\quad h_t + (uh)_x + (vh)_y = P \tag{2}$$

$u(x,y,t), v(x,y,t)$ are depth-averaged velocities, g gravity, $\zeta(x,y,t)$ is water level above the reference plane, c_f is dimensionless bottom friction, $h(x,y,t) = \zeta(x,y,t) - e(x,y)$ is water depth, $e(x,y)$ is the bottom elevation above the reference plane, and P is precipitation. Note that $h_t = \zeta_t$. These equations are to be completed with boundary/initial conditions to get a well-posed problem [10,21].

The momentum and the continuity equation are discretized by a conservative Finite Volume Method (FVM). The grid is a quadtree [18], with clustered DEM pixels as subgrid, Fig. 1. The variable grid size Δx is always a multiple of the pixel size δ. The continuity equation is mildly non-linear. Flooding and drying are an intrinsic part of this equation [5]. For the semi-implicit (or IMEX, IMplicit EXplicit) time integration, the $\theta-$method is applied combined with explicit wet cross sections. Momentum advection may be locally implicit, utilizing Carlson's method [13,16]. Based on this, the following numerical approximations are defined:

$$\left.\begin{aligned} &\frac{u^{k+1}-u^k}{\Delta t} + adv\left(u^{k+\max(0,1-1/c_u)}\right) + gD_x\zeta^{k+\theta} + \frac{u^{k+1}}{V^k}f^k = 0 \\ &\frac{v^{k+1}-v^k}{\Delta t} + adv\left(v^{k+\max(0,1-1/c_v)}\right) + gD_y\zeta^{k+\theta} + \frac{v^{k+1}}{V^k}f^k = 0 \end{aligned}\right\} \tag{3}$$

$$\frac{V(\zeta^{k+1}) - V(\zeta^k)}{\Delta t} + \sum_{\forall A\in V} A^k u_A^{k+\theta}\overrightarrow{n}_A = Q_p\ , u^{k+\theta} = (1-\theta)u^k + \theta u^{k+1} \tag{4}$$

$adv(u^{k+\max(0,1/c_u)}, v^{k+\max(0,1/c_v)})$ are advection approximations, $D_{x,y}\zeta$ denote numerical surface gradients, f is a bed friction function, A is a wet cross section, $\overrightarrow{n}_A$ is the outward normal of A, $V(\zeta)$ is a piecewise linear volume function, $Q_P = S_V{\cdot}P$ denotes precipitation, and $S_V = \sum_{\forall\delta\in V}\delta^2$ is the maximum surface of a grid cell when all pixels are wet. After elimination of u^{k+1} from (4), by substitution of (3) into (4), (4) is solved by preconditioned CG and Newton iteration.

The method is robust and capable of dealing with many different flow conditions, including rapidly varied flows [9,19,26].

3. Stability and accuracy problems of subgrids with thin water layers

This section starts with a heuristic explanation that flow in thin layers on slopes, imposes time step restrictions that are proportional to the pixel size due to the explicit implementation of cross sections in (4). Our argumentation is based on a simple example, example 0, see Fig. 2, where we consider flow per unit width.

The 1D steady state solution of (1, 2), for the boundary conditions as depicted by Fig. 2, is uniform flow with a uniform depth. The uniform depth is equal to the normal depth, e.g. [3], and is given by $\overline{h} = (c_f q^2/(g|e_x|))^{1/3}$. The uniform velocity $u = \sqrt{|e_x|/c_f}\sqrt{gh}$, where c_f is a constant bottom friction coefficient and $|e_x|$ the bottom slope. For $|e_x| = c_f = 0.1$, we get $h(x,\infty) = \overline{h} = 0.0634m$ and $u(x,\infty) = 0.7886\,m/s$. This is a critical slope with critical uniform flow [3]. Initially the bed is dry, when at $x = 0, t = 0$ the inflow starts. So the transient is a downhill flood wave. This transient solution can be derived, by approximation, from the kinematic wave equation [3,23]:

$$\frac{\partial h}{\partial t} + U'\frac{\partial h}{\partial x} = 0,\ h(x,t) = h_0(x - U't) \tag{5}$$

Obviously (5) is a transport equation with a general solution $h(x,t) = h_0(x - U't)$. Initially $h(x,0) = 0$. $h(x,t) = h_0(x - U't)$ implies that $h(x,t) = 0, t < x/U'$. From $h(x,\infty) = \overline{h}$ it follows that $h(x,t) = \overline{h}, t \geq x/U'$. The remaining question is the value of U'. For $T = L/U'$ the volume in the domain is $L\overline{h}$. We also have $T = L\overline{h}/q$, $q = u\overline{h} \rightarrow U' = u$

That is different from the river flow cases in [3,23] without dry bed. Here $U' = \alpha u$ and α depends on the bed friction formulation [3, page 147].

We compare computations for 3 different values of Δx: $\Delta x_1 = \delta$, $\Delta x_2 = 2\delta$, $\Delta x_3 = 4\delta$, $\delta = 1\ m$, Fig. 3. We established the maximum time step empirically. In all three cases, both the normal depth and the uniform velocity were computed without errors. In all cases the maximum time steps were $\Delta t = 1\ s$. Larger time steps were either oscillating or eventually unstable. For $\Delta x = 1$ that was slightly larger than the zero-stability condition of the appendix. But increasing Δx did

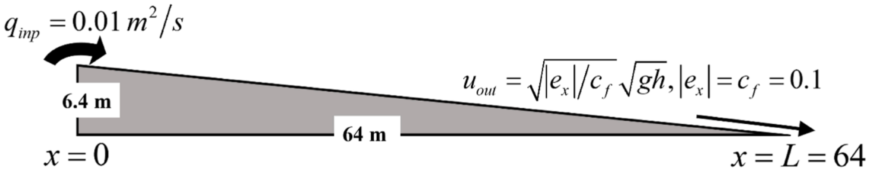

Fig. 2. Test case with a slope 1:10.

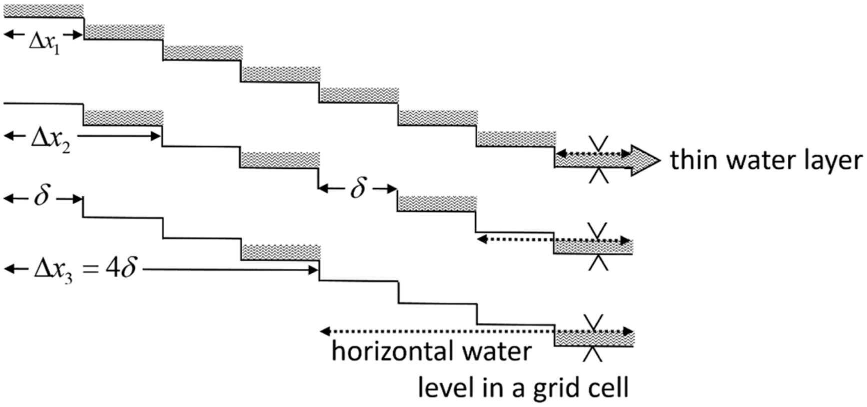

Fig. 3. Representation of a thin layer of water with a constant depth on a slope, for three grid sizes. For Δx_1 all pixels are wet, for Δx_2 50% of the pixels is wet and for Δx_3 only 25% of the pixels is wet.

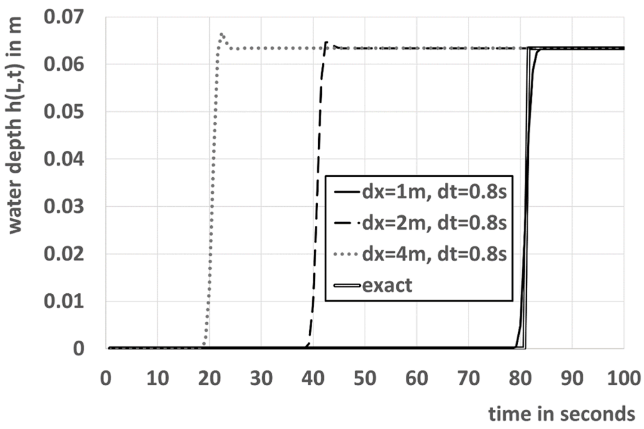

Fig. 4. 3 different numerical propagation speeds for $\Delta t = 0.8\,s$.

not change the stability constraints, while the transients were quite different, Fig. 4.

Fig. 4 shows the water depth $h(L,t)$ at $x = L = 64\ m$, the end of the slope. It takes a while until the water has reached the end point and the complete domain is at normal depth. But the larger the grid size, the earlier the entire domain is filled with water at normal depth. Fig. 3 explains this. There is simply less water in the domain $0 \le x \le L$ for a coarser grid with the same water depth, so the filling time is less. The height difference between pixels is 0.1 m, this is more than the normal depth of 0.0634 m. Because the water level in a grid cell is horizontal, only one pixel per grid cell is wet, regardless the grid size, Fig. 3.

The stability correlation with the pixel size δ rather than Δx is explained by the numerical approximation (4) applied to our simple problem. If we consider the equations per unit width, with thin layers, such that for each cell only the lowest pixels are wet, Fig. 3, then $V'(h) = \delta h$, (V' denotes volume per unit width) and (4) becomes:

$$\delta\frac{h_n^{k+1}-h_n^k}{\Delta t} + u_{n+1/2}^{k+1}h_n^k - u_{n-1/2}^{k+1}h_{n-1}^k = 0$$
$$\rightarrow \frac{h_n^{k+1}-h_n^k}{\Delta t} + \frac{u_{n+1/2}^{k+1}h_n^k - u_{n-1/2}^{k+1}h_{n-1}^k}{\delta} = 0 \tag{6}$$

We rewrite (6) as follows:

$$\frac{h_n^{k+1}-h_n^k}{\Delta t} + \frac{\frac{\Delta x}{\delta}u_{n+1/2}^{k+1}h_n^k - \frac{\Delta x}{\delta}u_{n-1/2}^{k+1}h_{n-1}^k}{\Delta x} = 0 \tag{7}$$

This is a consistent approximation of $\partial h/\partial t + \partial((\Delta x/\delta)uh)/\partial x = 0$. This implies that if $\Delta x = 4\delta$ the numerical propagation speed is four times the real one as is confirmed by Fig. 4. This also implies critical flow at an earlier stage and consequently instabilities pop up earlier.

In conclusion: The example of this section shows that thin layers with subgrids have issues with stability and accuracy. In the next two sections these problems will be tackled.

4. The volume correction equation

The basic idea of our Volume Correction Equation, VCE, is as follows: consider the following equation:

$$\frac{dV(h)}{dt} + Q(h) = 0,\ Q(h) = A(h)u(h) \tag{8}$$

Eq. (8) describes outflow from a volume $V(h)$ by a discharge $Q(h) = A(h)u(h)$. $V(h)$ and the cross section $A(h)$, are nonlinear functions of the water depth $h(t)$. The velocity $u(h)$ is linear. Note that $V_t = V_h h_t$. To integrate (8) we apply the following predictor-corrector procedure:

$$\text{Predictor}:\quad V_h\left(h^k\right)\frac{h^* - h^k}{\Delta t} + u(h^*)A\left(h^k\right) = 0 \tag{9a}$$

Eq. (9a) is linear and consistent with (8). It does not ensure positivity, nor conservation (i.e. the volume change is not equal to $\Delta t \times$ outflow). (9a) is followed by a corrector:

$$\text{Corrector:}\ \left\{\begin{array}{c} u^{k+1} = u(h^*),\quad \dfrac{V^{k+1}-V^k}{\Delta t} + Q^{k+1} = 0, \\ Q^{k+1} = u^{k+1}\left(\overbrace{A^k\dfrac{V^{k+1}}{\max\left(V^k,\varepsilon\right)}}^{\text{corrected cross section}}\right) \\ \rightarrow V^{k+1} = V^k\Bigg/\left(1+\Delta t\dfrac{u^{k+1}A^k}{\max\left(V^k,\varepsilon\right)}\right)\Bigg\rangle 0\,\forall\Delta t > 0, \\ V\left(h^{k+1}\right) = V^{k+1} \end{array}\right. \tag{9b}$$

The corrector (9b) is called VCE. Both conservation and positivity are secured. The value for u^{k+1} is from the predictor only. The nonlinear equation $V(h^{k+1}) = V^{k+1}$ is a single equation that can be solved with any iterative method. This approach is comparable with a treatment to ensure positive variables by [15, page 145]. In a trivial linear case, e.g. $V = hm^3$, $A = hm^2$, $u = 1\,m/s$, (9a) is overruled by (9b) and (9) becomes overall: $(h^{k+1} - h^k)/\Delta t + h^{k+1} = 0$, i.e. the implicit Euler rule [14].

For the general application of VCE to (3, 4) we consider the first iteration step for solving (3, 4), as the predictor. Then the generalized VCE formulation becomes:

$$\frac{V_n^{k+1}-V_n^k}{\Delta t}-\sum_{j=1,j\leftrightarrow n}^{n-1}Q_{j,n}^{k+1}+\sum_{l=n+1,n\leftrightarrow l}^{N}Q_{n,l}^{k+1}=Q_{P,n},\ n=1,\ldots,N \tag{10}$$

The expression $j\leftrightarrow n$ means that in a staggered grid V_j and V_n are adjacent nodes. $Q_{j,n}^{k+1}$ $(j<n\forall(j,n))$ is a discharge positively directed from V_j to V_n and $Q_{n,l}^{k+1}$ $(n<l\forall n,l)$ is defined accordingly. They are given by:

$$\left.\begin{aligned} Q_{j,n}^{k+1}&=\frac{\max\left(\widetilde{Q}_{j,n},0\right)}{V_j^k}V_j^{k+1}+\frac{\min\left(\widetilde{Q}_{j,n},0\right)}{V_n^k}V_n^{k+1},\ j\leftrightarrow n\\ Q_{n,l}^{k+1}&=\frac{\max\left(\widetilde{Q}_{n,l},0\right)}{V_n^k}V_n^{k+1}+\frac{\min\left(\widetilde{Q}_{n,l},0\right)}{V_l^k}V_l^{k+1},\ n\leftrightarrow l\end{aligned}\right\} \tag{11}$$

$\widetilde{Q}=u^{k+1}A^k$. (11) Implies that the correction factors V^{k+1}/V^k are from the upwind direction. Substitution of (11) into (10) gives the following linear equation:

$$\sum_{j=1}^{n-1}a_{n,j}V_n^{k+1}+a_{n,n}V_n^{k+1}+\sum_{l=n+1}^{N}a_{n,l}V_n^{k+1}=b_n,\ n=1,\ldots,N \tag{12}$$

The matrix coefficients $a_{n,j},\ j<n$ and $a_{n,l},\ l>n$ are given by:

$$\left.\begin{aligned} a_{n,j}&=-\max\left(\widetilde{Q}_{j,n},0\right)\Big/V_j^k,j<n\\ a_{n,l}&=\min\left(\widetilde{Q}_{n,l},0\right)/V_l^k,l>n\end{aligned}\right\} \tag{13}$$

If a pair of volume nodes (n,m) is not adjacent then $a_{n,m}=a_{m,n}=0$. If $a_{m,n}<0$ then $a_{n,m}=0\rightarrow a_{m,n}\le 0,\ \forall(m,\ n),\ m\neq n$. The coefficients $a_{n,n}$, b_n, are given by:

$$a_{n,n}=\frac{1}{\Delta t}+\sum_{m=1,N,\ m\neq n}\left|a_{n,m}\right|,\ b_n=\frac{V_n^k}{\Delta t}+Q_{P,n} \tag{14}$$

The value of ζ^{k+1} then follows directly from $V(\zeta^{k+1})=V^{k+1}$. This is a non-linear equation. But now, it is a collection of single mildly-non-linear equations, each with only one unknown. This excludes the use of matrix solvers in this final step. So matrix solvers are only applied for linear equations. The sparse matrix Eq. (12) will be solved by Gauss Seidel iterations [20]. From (14) it follows that (12) is a diagonally dominant (by column) equation which ensures convergence. Dominance by column or by row are each sufficient for convergence since a matrix and its transpose have the same eigenvalues. A Gauss Seidel iteration step to solve (12) is:

$$V_n^{k+1,i}=\frac{-\sum_{j=1}^{n-1}a_{n,j}V_n^{k+1,i}-\sum_{l=n+1}^{N}a_{n,l}V_n^{k+1,i-1}+b_n}{a_{n,n}},\ n=1,\ldots,N,\ i\ge 1 \tag{15}$$

$V_n^{k+1,0}=V_n^k$, (15) is repeated for $i=1,\ldots$until convergence. If $V_n^k>0$, $\forall n$ then $a_{n,n}>0,\forall n$ and $a_{m,n}<0,\forall m\neq n$. From this it follows that $V_n^{k+1,i}>0,\forall n,i$ and therefore also $V_n^{k+1}>0,\forall n$. So VCE will guarantee positive volumes. If we assume $\widetilde{Q}_{m,n}\ge 0,\ \forall(m,n)$ then $a_{n,m}=0,\ m>n$. This implies for (15):

$$V_n^{k+1,i}=\frac{-\sum_{j=1}^{n-1}a_{n,j}V_n^{k+1,i}+b_n}{a_{n,n}},\ n=1,\ldots,N,\ i\ge 1$$

This means that the exact solution of (12) is found in one sweep. If we reverse the sweep direction of (15) we get:

$$V_n^{k+1,i}=\frac{-\sum_{j=1}^{n-1}a_{n,j}V_n^{k+1,i-1}-\sum_{l=n+1}^{N}a_{n,l}V_n^{k+1,i}+b_n}{a_{n,n}},\ n=N,\ldots,1,\ i\ge 1 \tag{16}$$

Here the exact solution is found in one sweep if $\widetilde{Q}_{m,n}\le 0,\ \forall(m,n)$. Normally Gauss Seidel proceeds with alternating sweep directions.

VCE will stabilize the problem of Section 3. Much larger time steps than the empirically established limit of one second are feasible, but will not improve the accuracy as Fig. 5 illustrates. The steady state however is still exact only the transient has large errors for $\Delta x=2\ m$ and $\Delta x=4\ m$.

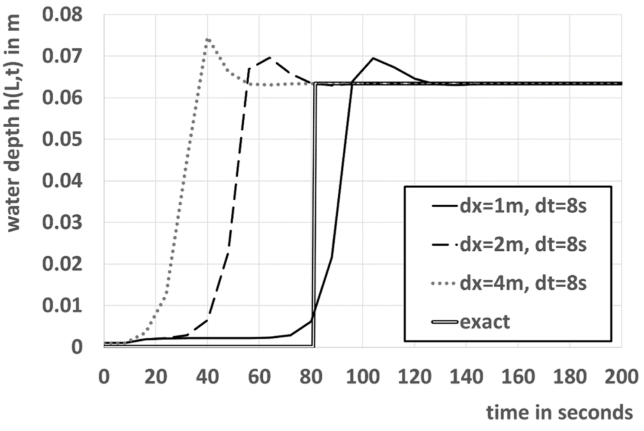

Fig. 5. Stabilizing effect of VCE, $\Delta t=8\ s$, $\Delta x=1,2,4\,m$.

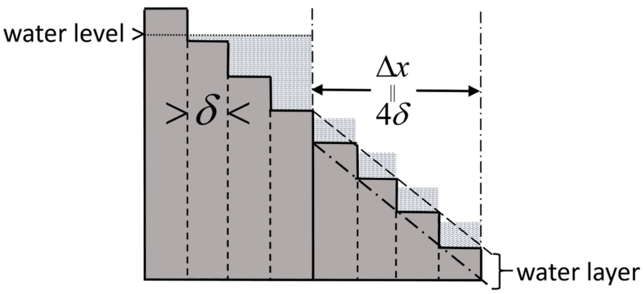

Fig. 6. Horizontal water level vs sloping water layer.

5. Thin layer calculation on slopes

The basic assumption of the subgrid method is a horizontal water level within a grid cell, Fig. 6. For rainfall-runoff on slopes, this assumption is not practical and will lead to large errors, if the cells are not sufficiently small. Here accurate solutions may be obtained only if the cell size is just as large as a pixel, as our example 0 in Section 3 has shown. This is contrary to the general experience of accurate solutions with subgrid methods. An explanation is given in Fig. 6. Here the same amount of water is divided in two different ways: on the left side the water level is horizontal, on the right side the water depth is constant. If it concerns a slope with overland flow, then the horizontal assumption is wrong and will underestimate friction and overestimate cross-sections. However, when this is the edge of a pond, with almost still water or channel flow in a direction normal to the figure, then it is right. Both volumes are given by:

$$V(\zeta)=\delta^2\sum_{\forall i\in V}(\zeta-e_i)\Lambda(\zeta-e_i),\ \Lambda(h)=\begin{cases}1\Leftrightarrow h>0\\ 0\Leftrightarrow h\le 0\end{cases} \tag{17}$$

For the right side of Fig. 6 we define the *thin layer depth,* h_Γ:

$$h_\Gamma=V/S_V \tag{18}$$

Eq. (18) prescribes the *thin layer calculation,* TLC. The depth h_Γ is used both for cross-sections and bed friction. We compute the wet cross-section A_Γ between two cells V_- and V_+ as follows:

$$A_\Gamma=\begin{cases}h_{\Gamma,-}\cdot w_{-\cap+} & u>0\\ \max\left(h_{\Gamma,-},h_{\Gamma,+}\right)\cdot w_{-\cap+} & u=0\\ h_{\Gamma,+}\cdot w_{-\cap+} & u<0\end{cases} \tag{19}$$

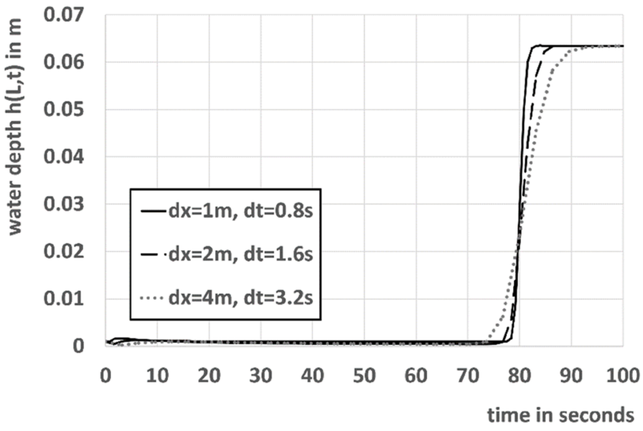

Fig. 7. TLC for 3 different values of $\Delta t$and Δx.

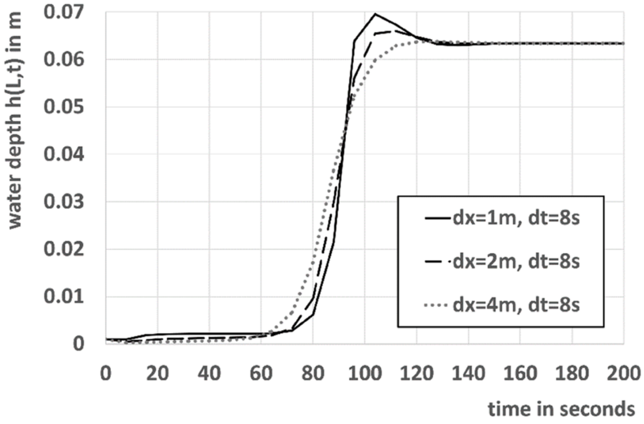

Fig. 8. VCE and TLC, $\Delta t = 8s,\ \Delta x = 1,2,4\,m$.

Here $w_{-\cap+}$ is the maximum width of the mutual face between V_- and V_+, and u is the velocity at the interface. The “normal” cross-section, A_H based on the horizontal water level assumption within a computational cell, is given by:

$$A_H = \begin{cases} \delta \sum\limits_{j \in face} \Lambda(\zeta_- - e_j)\,, & u > 0 \\ \delta \sum\limits_{j \in face} \Lambda(\max(\zeta_-, \zeta_+) - e_j) & u = 0 \\ \delta \sum\limits_{j \in face} \Lambda(\zeta_+ - e_j), & u < 0 \end{cases} \tag{20}$$

Both approximations are consistent. If a grid cell has the size of just one pixel then (19) and (20) are identical. I we apply (19) we get the phase speed right for all values of Δx, Fig. 7. The stability condition is now proportional to Δx rather than δ. Also example 1 in Section 6, shows remarkable improvement due to (19).

In combination with VCE the time steps can be increased, Fig. 8.

In practical applications the combinations VCE and TLC will allow much larger time steps.

The actual cross section A is given by:

$$A = \begin{cases} \min(A_H, A_\Gamma), & h_{\min,+\cap-} \le 0 \\ A_H, & h_{\min,+\cap-} > 0 \end{cases} \tag{21}$$

$h_{\min,+\cap-} = \min(\zeta_-, \zeta_+) - e_{\min,-\cap+}$, $e_{\min,-\cap+}$ is the lowest pixel at the interface of $V_-$and V_+. Eq. (21) implies that if there is flow through the interface between two cells, while the downstream water level is below the lowest pixel at the interface, then TLC is applied. The combination of VCE and TLC is very effective as we will show in the next section.

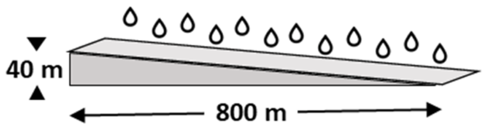

Fig. 9. bottom slope example 1.

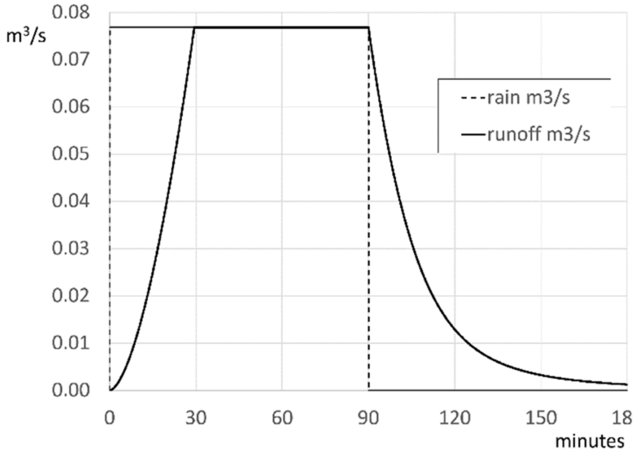

Fig. 10. analytical hydrograph.

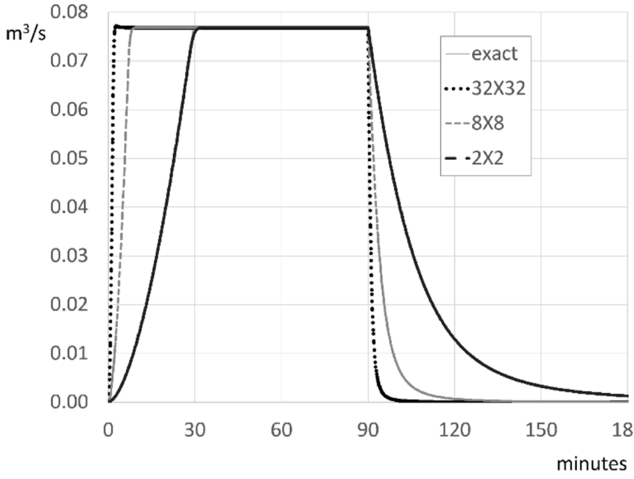

Fig. 11. numerical runoff without TLC.

6. Examples

In this section we describe two examples. For all these examples the bed friction is based on Manning’s N:

$$c_f = gN^2 \,/\, h^{1/3} \tag{22}$$

6.1. Example 1

The first example is based on a well-known analytical solution [2] for the kinematic wave equation that is often applied [8]. Fig. 9 shows the bathymetry.

The slope of 5% has a length of 800 m and a width of 32 m. This bathymetry is represented by a raster of 800 × 32 pixels of 1 × 1 m^2. Manning’s $N = 0.015$. The precipitation is 10.8 mm/hr for 90 min. The analytical hydrograph is given by Fig. 10. Time is given in minutes.

Fig. 11 shows results without TLC, for a coarse grid cell size of 32 × 32, 8 × 8 and 2 × 2 pixels. The runoff results of 32 × 32 and even of 8 ×

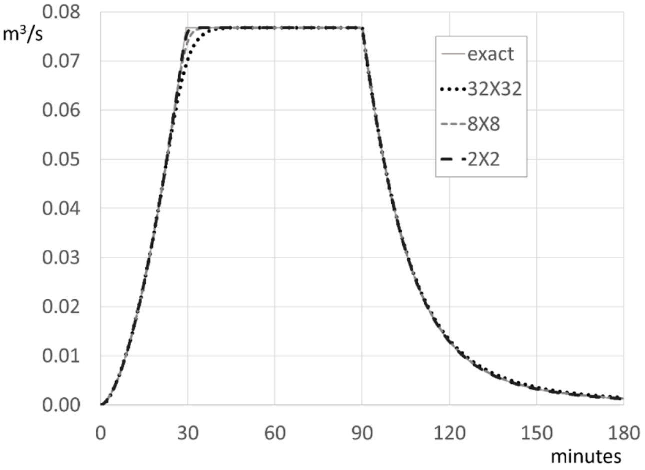

Fig. 12. numerical runoff with TLC.

8 pixels are way too fast, while the 2 × 2 case is perfect. So, without TLC the subgrid method is far from accurate for runoff on a simple flat slope. Computations without subgrid will even be better. Fig. 12 shows results for the same grids but now with TLC. The improvement is remarkable. A coarse grid cell size of 32 × 32 pixels is already fairly precise.

6.2. *Example 2*

To demonstrate the effect of VCE/TLC for practical applications, we have constructed a second example. We have composed a bathymetry derived from a real DEM with realistic slopes, Fig. 13. This example has overland flow, ponding and channel flow. The method will switch between TLC and the normal subgrid approach, based on (21). This is where the combination of VCE and TLC shows not only to be accurate but also to be robust and very stable.

The bathymetry on Fig. 13, of 7×6 km, represents a small village, nearby hills that range from 60 m to 200 m above chart datum. (Amsterdam Ordnance Datum, NAP). The DEM's pixel size is 1 × 1 m^2. The evenly colored gray area is outside the computational domain. The subgrid domain contains 23.2 million pixels. A constant Manning value of $N = 0.04$ has been applied. This area will receive a tropical rainstorm with a peak of 17.7 cm/hr. See Fig. 14 for a hydrograph of 6 h. In fact such a heavy rainstorm of short duration is not (yet?) realistic in this area. The rainstorm causes a flash flood, with inundated areas, Figs. 15 and 16. In the 2 × 2 case, the effect of TLC is negligible as we have shown in example 1. We conclude from Fig. 17 that for a grid cell of 16 ×

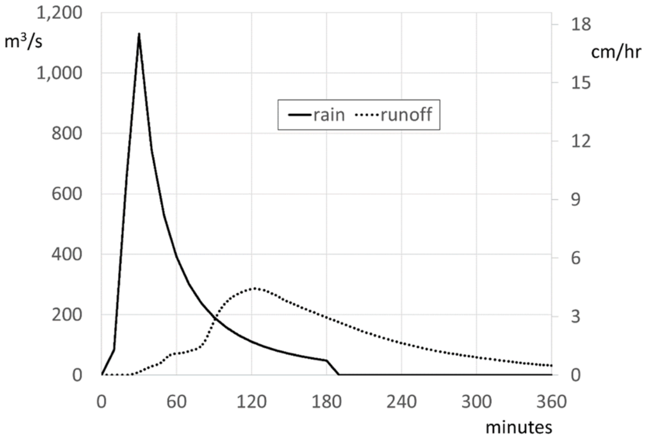

Fig. 14. Hydrograph example 2, time in minutes, rain and runoff in m^3/s(left) or cm/hr (right).

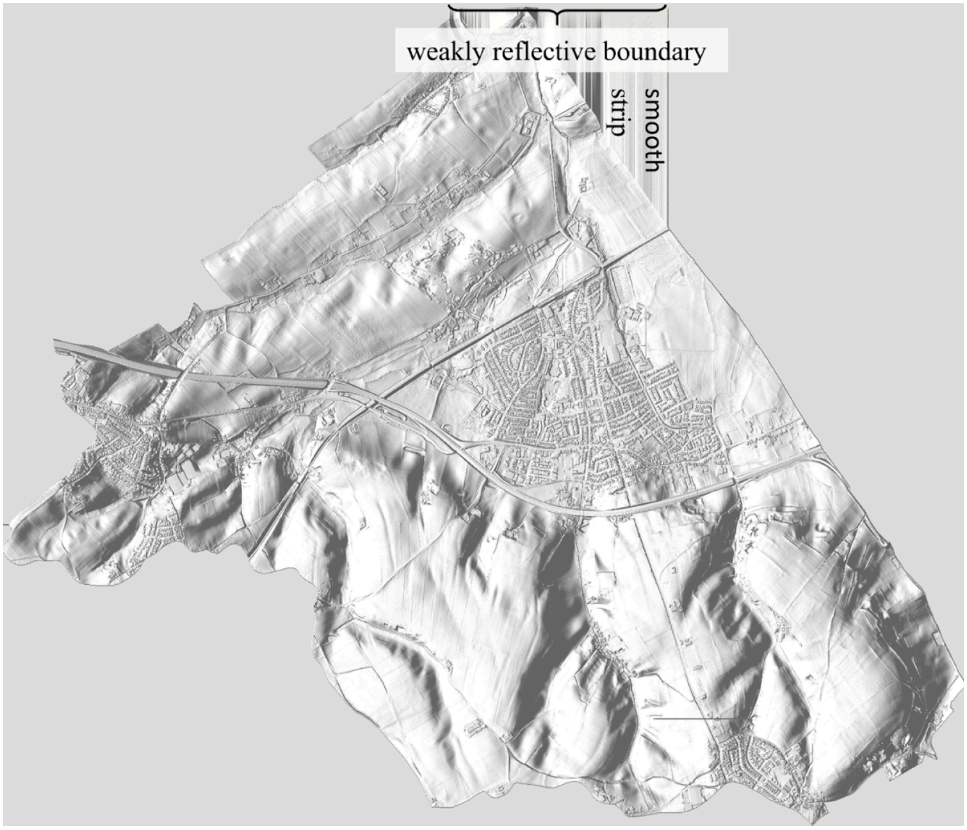

Fig. 13. Bathymetry example 2.

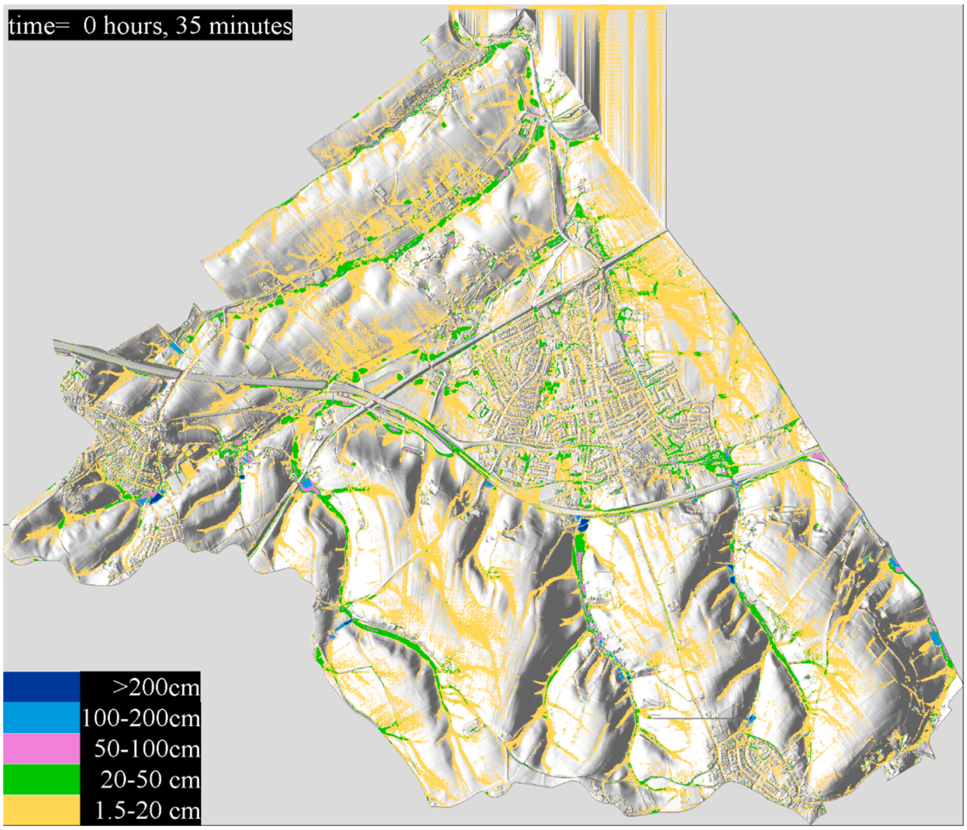

Fig. 15. inundation depths after 35 min, shallow streams of water are all over in the folds of the hills, no serious ponding yet.

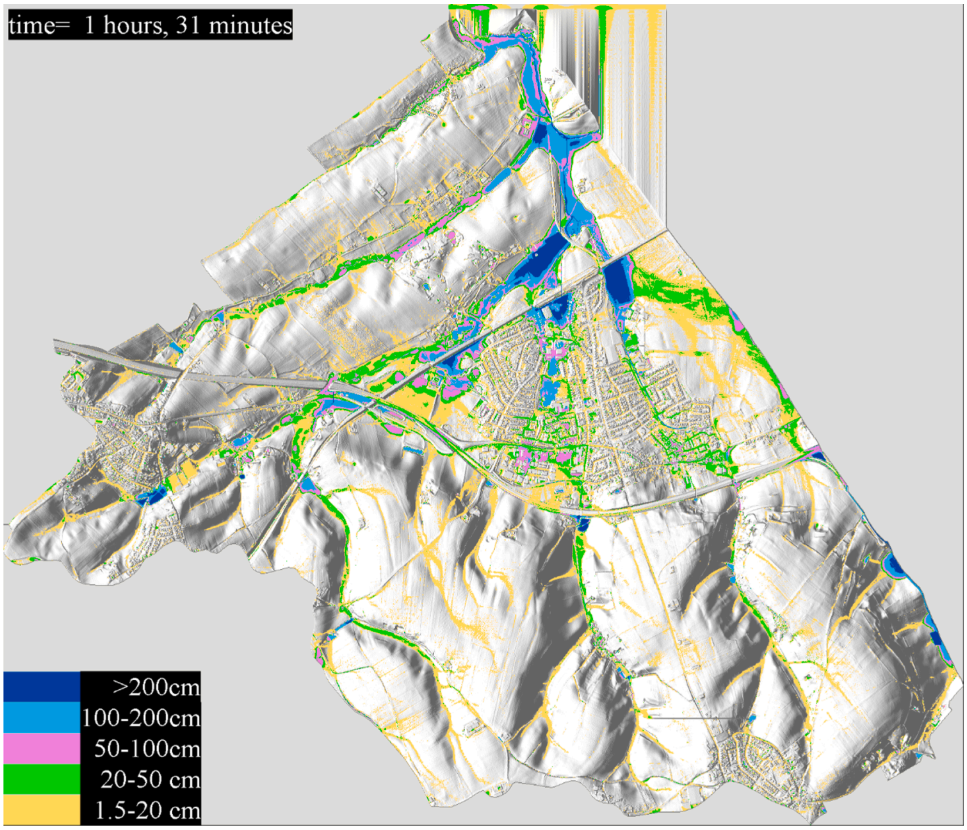

Fig. 16. inundation depths after 91 min, the rain storm is almost over, some small streams are still present, ponding occurred even with depths $>$ 2 m.

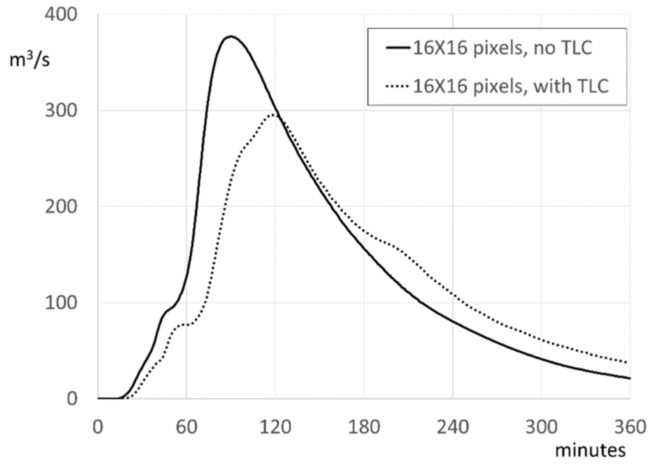

Fig. 17. runoff comparison without and with TLC, differences are significant.

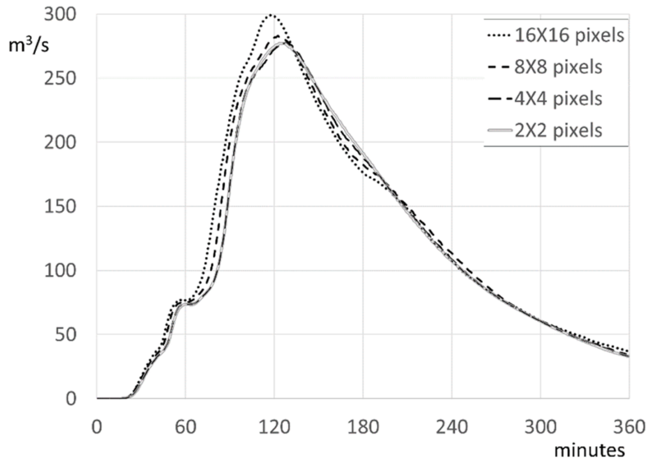

Fig. 18. runoff convergence test with TLC.

16 pixels, the differences are prominent. In Fig. 18 we explore the effect of refining grid cells. This may have a significant effect [25]. Here we give results for grid cells of 16×16, 8×8, 4×4 and 2×2 pixels. The results of 8×8, 4×4 and 2×2 pixels are close.

For each of the hydrographs in Fig. 18, the table below contains the following data: grid size, number of grid points, Δt, wall time, percentage of wall time due to the CG matrix solver, percentage of wall time by the Gauss Seidel solver of the VCE and the accumulated mass balance error (MBE). TLC is always applied. If %VCE=0 then VCE is not applied.

Table of computational effort with a 12 core CPU (AMD ryzen-9 5900X). The FORTRAN code was parallelized with openMP directives.

grid size	grid points	Δt	wall time	% CG	% VCE	MBE
16×16 pixels	91,461	5 s	93 s	66%	0	$1 \times 10^{-6} m^3$
16×16 pixels	91,461	5 s	77 s	35%	27%	$7 \times 10^{-8} m^3$
16×16 pixels	91,461	30 s	24 s	58%	23%	$6 \times 10^{-8} m^3$
8 × 8 pixels	363,706	2.5 s	16.6 min	71%	0	$6 \times 10^{-7} m^3$
8 × 8 pixels	363,706	2.5 s	10.7 min	32%	24%	$1.5 \times 10^{-7} m^3$
8 × 8 pixels	363,706	30 s	2.5 min	64%	20%	$1 \times 10^{-7} m^3$
4 × 4 pixels	1,450,560	2.5 s	4 hrs. 8 min.	92%	0	$1.5 \times 10^{-7} m^3$
4 × 4 pixels	1,450,560	2.5 s	1 hr 25 min.	65%	18%	$1.5 \times 10^{-7} m^3$
4 × 4 pixels	1,450,560	30 s	50 min	86%	11%	$5 \times 10^{-7} m^3$
2 × 2 pixels	5,794,459	15 s	9 hrs. 54 min.	86%	12%	$1 \times 10^{-6} m^3$

A pre-conditioned CG algorithm solves the semi-implicit predictor. The VCE is solved by Gauss-Seidel. CG consumes the major part of the computation time in particular for large applications. The time steps in the table with VCE only, were unstable without VCE. This proves the important benefit of VCE, with only 11% to 27% of computational overhead. The table also shows that for the same time steps, the computations with VCE are always faster. Comparison is only possible for small time steps due to instability without VCE. For 4×4 pixels the difference is a factor 3. All the mass balance errors are negligible, compared to an overall mass of $O(2 \times 10^6)$ m^3. With VCE local Courant numbers may rise above 50. The results of the small and large time steps did not show significant differences for the hydrograph for equal values of the grid size. The overall CPU time does not decrease linearly with an increasing time step. This is due to the CG solver that requires more iterations per time step. There is an optimum, where above further increase of the time step will also increase the overall computational effort.

From a practical point of view, the differences of the results between 8×8, 4×4 and 2×2 pixels are small. The results of 16×16 could still be useful for rapid assessment applications. The computation times however, are extremely different, ranging from 24 s to almost 10 h. Combined with adaptive quadtrees, very efficient flood mapping is possible for large catchments, even with the modest hardware used for this paper.

7. Concluding remarks

The quest for ever more efficient and accurate SWE-solvers is a never ending story. The subgrid method [5] has contributed a lot. But, as this paper shows, there are drawbacks on steep slopes with friction dominated flow. Here the explicit implementation of the cross sections yields time step restrictions proportional to the pixel size as this paper shows. Due to an erroneous phase speed proportional to $u\Delta x/\delta$, time steps were required to be impossibly small, because of artificially large Froude number. In particular during the spin-up of a simulation this will occur. This is a classic example of a stiff problem, the time steps required for stability are much smaller than the time steps required for accuracy [14]. Euler implicit time integration is the most robust approach to deal with stiffness, but to solve this in one step with only implicit approximations of each term of (1,2) will be an extremely demanding computational effort. So, there will always be the need for an efficient compromise between explicit and implicit terms. This may differ, depending on the particular application. VCE combined with TLC seems to be such a proper balance for rainfall runoff on slopes. The maximum grid size for accurate solutions of example 2 is 8 pixels, although 16 was still reasonable. That is a bit disappointing. This is probably due to ignoring important obstacles for large values of Δx. This may be improved by methods, as mentioned in [4,7,11], or by adaptive quadtrees. Finally we arrive at the following conclusions:

- The strictly conservative VCE improves robustness and efficiency, due to improved stability and because of bypassing the repetitive application of matrix solvers as part of Newton iteration.
- TLC is imperative for accurate overland flow on slopes, with coarse grids and sloping subgrids.
- The combination of VCE and TLC (and detailed DEM's), enables efficient flash flood mapping, for large catchments with hills, provided that flow blocking obstacles are not overlooked by the grid resolution.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data will be made available on request.

Acknowledgment

The author is greatful to the reviewers for their valuable and inspiring comments.

Appendix, Stability of semi-implicit SWE time integration

To show that the explicit part of the semi-implicit method will limit the time step, at least for critical flows, we apply Neumann stability analysis for 1D SWE, given by:

$$u_t + uu_x + g\zeta_x + \frac{c_f}{h}u|u| = 0 \tag{A-1}$$

$$h_t + (uh)_x = \zeta_t + u\zeta_x - ue_x + hu_x = 0 \tag{A-2}$$

We apply, while disregarding subgrids, the semi-implicit method with a staggered grid to the frozen-coefficient formulation of (A-1, 2) with Θ=1 and *U>0*:

$$\left.\begin{array}{r} \dfrac{u^{k+1}_{m+1/2} - u^{k}_{m+1/2}}{\Delta t} + U\dfrac{u^{k}_{m+1/2} - u^{k}_{m-1/2}}{\Delta x} + g\dfrac{\zeta^{k+1}_{m+1} - \zeta^{k+1}_{m}}{\Delta x} + \dfrac{c_f U}{H}u^{k+1} = 0 \\ \dfrac{\zeta^{k+1}_{m} - \zeta^{k}_{m}}{\Delta t} + U\dfrac{\zeta^{k}_{m} - \zeta^{k}_{m-1}}{\Delta x} + H\dfrac{u^{k+1}_{m+1/2} - u^{k+1}_{m-1/2}}{\Delta x} - Ue_x = 0 \end{array}\right\} \tag{A-3}$$

We rewrite (A-3):

$$u^{k+1}_{m+1/2} + rg\left(\zeta^{k+1}_{m+1} - \zeta^{k+1}_{m}\right) + \Delta t\frac{c_f U}{H}u^{k+1} = u^{k}_{m+1/2} - rU\left(u^{k}_{m+1/2} - u^{k}_{m-1/2}\right)$$

$$\zeta^{k+1}_{m} + rH\left(u^{k+1}_{m+1/2} - u^{k+1}_{m-1/2}\right) - \Delta t U e_x = \zeta^{k}_{m} - rU\left(\zeta^{k}_{m+1} - \zeta^{k}_{m}\right)$$

Here *r=Δt/Δx*. To study zero stability by Von Neumann's analysis, that implies $\Delta t \to 0,\ \Delta x \to 0$ while *r* is constant, we get:

$$\begin{bmatrix} 1 & r\sqrt{gH}2i\sin\left(\frac{1}{2}\varphi\right) \\ r\sqrt{gH}2i\sin\left(\frac{1}{2}\varphi\right) & 1 \end{bmatrix}\begin{bmatrix} \hat{z}^{k+1} \\ \hat{u}^{k+1} \end{bmatrix} = \begin{bmatrix} 1 - rU(1 - \cos\varphi + i\sin\varphi) & 0 \\ 0 & 1 - rU(1 - \cos\varphi + i\sin\varphi) \end{bmatrix}\begin{bmatrix} \hat{z}^{k} \\ \hat{u}^{k} \end{bmatrix} \tag{A-4}$$

$\hat{z} = \sqrt{g/H}\hat{\zeta}$ and $\varphi = i\kappa\Delta x$.

For zero stability or convergence we study Δt→0 while r is kept constant. In that case all O(Δt) terms disappear [16]. We diagonalise (A-4) to get:

$$\left(1 + r_C 2i\sin\left(\frac{1}{2}\varphi\right)\right)\left(\hat{u}^{k+1} + \hat{z}^{k+1}\right) = \left(1 - r_U(1 - \cos\varphi + i\sin\varphi)\right)\left(\hat{u}^{k} + \hat{z}^{k}\right)$$

$$\left(1 - r_C 2i\sin\left(\frac{1}{2}\varphi\right)\right)\left(\hat{u}^{k+1} - \hat{z}^{k+1}\right) = \left(1 - r_U(1 - \cos\varphi + i\sin\varphi)\right)\left(\hat{u}^{k} - \hat{z}^{k}\right)$$

$r_C = \frac{\Delta t}{\Delta x}\sqrt{gH}, r_U = \frac{\Delta t}{\Delta x}U$. The amplication factor *A* is given by:

$$A = \frac{1 - r_U(1 - \cos\varphi + i\sin\varphi)}{1 \pm r_C 2i\sin\left(\frac{1}{2}\varphi\right)}$$

$|A| \leq 1$ is a necessary (but not sufficient!) condition for stability. This is satisfied iff:

$$\left(1 - \Phi^2\right)r_U < 1 \tag{A-5}$$

$\Phi = r_C/r_U$. Φ is the inverse of the Froude number. If $\Phi > 1$, i.e. for sub-critical flows, then (A-5) bears no time step restrictions, but for $\Phi = 0$, i.e.: for super critical flows or in case of small depths, we get $r_U < 1$, i.e. the velocity Courant condition, regardless of slopes or drying. On slopes, thin water layers may have swift flows, such that (A-5) is a significant limitation. Bear in mind that for 2D SWE $r_U = \sqrt{2}U\Delta t/\Delta x$.

References

[1] Afifi Z, Chu HJ, Kuo YL, Yung-Chia HKW, Ali MZ. Residential flood loss assessment and risk mapping from high-resolution simulation. Water 2019;11:751. https://doi.org/10.3390/w11040751 (Basel).

[2] Baiamonte G, Agnese C. An analytical solution of kinematic wave equations for overland flow. J Agric Eng 2010;1:41–9.

[3] Battjes JA, Labeur RJ. Unsteady flow in open channels. Cambridge University Press; 2017. ISBN 978-1-107-15029-4.

[4] Begmohammadi A, Wirasaet D, Silver Z, Bolster D, Kennedy AB, Dietrich JC. Subgrid surface connectivity for storm surge modeling. Adv Water Resour 2021; 153:103939. https://doi.org/10.1016/j.advwatres.2021.103939.

[5] Casulli V. A high-resolution wetting and drying algorithm for free-surface hydrodynamics. Int J Numer Meth Fluids 2009;60:391–408. https://doi.org/10.1002/fld.1896.

[6] Casulli V, Stelling GS. Semi-implicit sub-grid modelling of three-dimensional free-surface flows. Int J Numer Meth Fluids 2011;67(4):441–9. https://doi.org/10.1002/fld.2361.

[7] Casulli V. Computational grid, subgrid, and pixels. Int J Numer Meth Fluids 2019; 90:140–55. https://doi.org/10.1002/fld.4715.

[8] Di Giammarco P, Todini E, Lamberti P. A conservative finite element approach to overland flow: the control finite volume element formulation. J Hydrol (Amst) 1996;175:267–91.

[9] Dumbser M, Casulli V. A staggered semi-implicit spectral discontinuous Galerkin scheme for the shallow water equations. Appl Math Comput 2013;219:8057–77. https://doi.org/10.1016/j.amc.2013.02.041.

[10] Engquist B, Majda A. Absorbing boundary conditions for the numerical simulation of waves. Math Comput 1977;31(139):629–51.

[11] Hodges BR. Representing hydrodynamically important blocking features in coastal or riverine lidar topography. Nat. Hazards Earth Syst Sci 2015;15:1011–23. https://doi.org/10.5194/nhess-15-1011-2015.

[12] Kennedy AB, Wirasaet D, Begmohammadi A, Sherman T, Bolster D, Dietrich JC. Subgrid theory for storm surge modeling. Ocean Model 2019;144. https://doi.org/10.1016/j.ocemod.2019.101491.

[13] Kramer SC, Stelling GS. A conservative unstructured scheme for rapidly varied flows. Int J Numer Meth Fluids 2008;58:183–212. https://doi.org/10.1002/fld.1722.

[14] Lambert JD. Numerical methods for ordinary differential systems. John Wiley &Sons; 1991. ISBN 0 471 92990 5.

[15] Patankar SV. Numerical heat transfer and fluid flow. Hemisphere Publishing Corporation; 1980. ISBN 0-07-048740-5.

[16] Richtmyer RD, Morton KW. Difference methods for initial value problems. Interscience Publishers; 1967. ISBN 0 470 72040 9.

[17] Ruheili AA, Dahm R, Radke J. Wadi flood impact assessment of the 2002 cyclonic storm in Dhofar, Oman under present and future sea level conditions. J. Arid Environ. 2019;165:73–80. https://doi.org/10.1016/j.jaridenv.2019.04.002.

[18] Stelling GS. Quadtree flood simulations with sub-grid digital elevation models. Water Manag 2012;(WM10):567–80. https://doi.org/10.1680/wama.12.00018.

[19] Stelling GS, Duinmeijer SPA. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. Int J Numer Methods Fluids 2003;43 (12):1329–54. https://doi.org/10.1002/d.537.

[20] Varga RS. Matrix iterative analysis, revised and extended edition. Springer; 2000. https://doi.org/10.1007/978-3-642-05156-2. e-ISBN 978-3-642-05156-2.

[21] Verboom GK, Slob A. Weakly-reflective boundary conditions for two dimensional shallow water flow problems. Adv Water Resour 1984;7:192–7.

[22] Viero DP, Valipour M. Modeling anisotropy in free-surface overland and shallow inundation flows. Adv Water Resour 2017;104(2017):1–14. https://doi.org/10.1016/j.advwatres.2017.03.007.

[23] Vreugdenhil CB. Computational hydraulics, an introduction, page 30. Springer Verlag; 1989. ISBN 0-387-50606-3.

[24] Wang Harry V, Loftis JD, Liu Z, Forrest D, Zhang J. The storm surge and sub-grid inundation modeling in New York City during hurricane sandy. J Mar Sci Eng 2014;2:226–46. https://doi.org/10.3390/jmse2010226.

[25] Hsu YC, Prinsen G, Bouaziz L, Lina YJ, Dahm R. An Investigation of DEM resolution influence on flood inundation simulation, 12th International Conference on Hydroinformatics. HIC 2016,Proced Eng 2016;154:826–34. https://doi.org/10.1016/j.proeng.2016.07.435.

[26] Zijlema M. The role of the Rankine-Hugoniot relations in staggered finite difference schemes for the shallow water equations. Comput Fluids 2019;192. https://doi.org/10.1016/j.compfluid. 2019. 104274.

[27] Liu Z. Operational flood forecasting and warning under the changing environment in China. Proc. IAHS 2020;383:223–8. https://doi.org/10.5194/piahs-383-223-2020.