V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

IMPLICIT SIMULATIONS OF SHALLOW-WATER EQUATIONS WITH MOBILE BED

Marco Bilanceri*, Imad Elmahi†, Hervé Guillard††, Maria Vittoria Salvetti* and François Beux‡

* Dip. Ingegneria Aerospaziale, Università di Pisa
Via G. Caruso 8, 56122 Pisa (Italy)
e-mail: {m.bilanceri, mv.salvetti}@ing.unipi.it
† EMCS, Ensa, Oujda
Complexe Universitaire, BP 669, 60000 Oujda (Marocco)
e-mail: ielmahi@ensa.ump.ma
†† INRIA
BP 93, 06902 Sophia Antipolis Cedex (France)
e-mail: Herve.Guillard@sophia.inria.fr
‡ Alta S.p.A.
Via Gherardesca, 5 - 56121 Ospedaletto, Pisa (Italy)
e-mail: f.beux@alta-space.com

Key words: Shallow water, mobile bed, implicit time advancing, automatic differentiation

Abstract. *A numerical methodology for the simulation of sediment transport is considered. The model is based on the shallow-water equations coupled with a sediment transport equation for the morphodynamic, namely the Exner equation and the Grass model. The aim of the present paper is to investigate the behavior of implicit linearized schemes in this context. First, 1D problems are considered and then the approach is extended to 2D ones. The equations are discretized in space through a finite-volume approach and second-order accuracy is obtained through MUSCL reconstruction. The implicit linearized scheme is derived by computing the Jacobian of the finite-volume fluxes using an automatic differentiation tool. Second order accuracy in time is obtained through a two step Runge-Kutta method for the explicit scheme and one or two iterations of Defect Correction for the implicit one. The different time-advancing schemes are compared, both in terms of accuracy and efficiency with different types of flow/bed interactions, namely strong, intermediate and weak interaction. It is shown that, while the results are in good agreement for both the explicit and implicit schemes, the implicit scheme is cost-effective for intermediate and weak flow/bed interactions and, as a consequence, it turns out to be a good candidate to simulate flows with sediment transport.*

1 Introduction

The design and validation of numerical methods for the simulation of bedload sediment transport processes caused by the movement of a fluid in contact with the sediment layer has a significant interest for environmental and engineering problems. A few examples of such problems are beach profile changes due to severe climate waves, seabed response to dredging procedures or imposed structures, harbor siltation or transport in gravel-bed rivers.

The hydrodynamics part is usually modeled through the classical shallow-water equations coupled with an additional equation modeling the morphodynamical component. This last equation is usually a continuity or Exner equation, expressing the conservation of the sediment volume, in which the solid transport discharge is provided by a closure model. Many different models of solid transport discharge are available in the literature (see, e.g., [1] for a review). As a first step, the Grass equation [2] is considered herein, which is one of the most popular and simple models.

A huge amount of work has been done in the last decades to develop numerical methods for the simulation of the previous system of equations (see, e.g., the references in [1, 3, 4]). The treatment of source terms and of the bed-load fluxes has received the largest attention. Indeed, a well known problem is that shallow water equations on non-flat topography have steady-state solutions in which the flux gradients are non-zero but are exactly balanced by the source terms. Standard numerical methods for the discretization of conservation laws may fail in correctly reproducing this balance (C-property, see e.g. [5]).

On the other hand, in shallow-water problems, time advancing has received much less attention and it is usually carried out by explicit schemes. The focus of the present paper is on the comparison between explicit and implicit schemes in the simulation of coupled shallow-water equations and sediment transport. Indeed, if the interaction of the water flow with the mobile bed is weak, the characteristic time scales of the flow and of the sediment transport can be very different introducing time stiffness in the global problem. For these cases, the stability properties of explicit schemes may significantly be deteriorated and, hence, it can be advantageous to use implicit schemes. Implicit schemes might also be useful if morphodynamic models more complex than the Exner/Grass one, which lead to a more stiff evolution of the bed (see e.g. [1]), are used. On the other hand, since the considered problems are unsteady, attention must be paid for implicit schemes in the choice of the time step. Indeed, a too large time step could deteriorate the accuracy of the results and one issue is to investigate whether and for which conditions the use of implicit schemes is really convenient from a computational viewpoint. A first investigation of this issue is provided in the present paper for 1D and 2D sediment-transport problems, involving different rates of bedload/water-flow interaction. Another difficulty with implicit schemes is that, in order to avoid the solution of a nonlinear system at each time step, the numerical fluxes must be linearized in time and this is classically done via differentiation by computing the Jacobian of the fluxes with respect to the flow variables.

Nevertheless, it is not always possible nor convenient to exactly compute the Jacobian matrices, because it is not unusual to have some lack of differentiability of the numerical flux functions or the computation may be complex for some schemes, as e.g. those involving projector-corrector stages. In order to overcome these difficulties, we use an automatic differentiation tool (Tapenade, [6], http://www-sop.inria.fr/tropics/). Our starting point was a numerical scheme developed and validated for the numerical simulation of sediment transport problems, in which the equations are discretized in space through a finite-volume approach and a non-homogeneous Riemann solver on unstructured grids [4]. The original version of the scheme was explicit, involving two stages (predictor and corrector); MUSCL reconstruction and a two-step Runge-Kutta time-integration method were used to obtain second-order accuracy in space and time respectively. Starting from this scheme, an implicit version is derived herein by computing the Jacobian matrices of the first-order accurate numerical fluxes by the previously mentioned automatic differentiation tool. A defect-correction approach [7], which consists in iteratively solving linear systems involving the 1st-order flux Jacobians, is finally used to obtain second-order accuracy (both in time and space) at limited computational costs. Note that automatic differentiation together with the defect-correct approach allow the numerical method to be easily adapted to changes in the physical model, such as for instance the use of different models for the solid transport discharge.

As previously mentioned, the accuracy and efficiency of the implicit and explicit versions of the numerical method are compared in 1D and 2D coupled water-flow/sediment-transport problems.

2 Physical Model

The physical model used in this work consists in the well known shallow-water equations coupled with an additional equation to describe the transport of sediment. Neglecting the wind effects, Coriolis forces and friction losses the 2D shallow water equations may be written in the following conservative form:

$$\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0$$

$$\frac{\partial hu}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(huv) = -gh\frac{\partial Z}{\partial x} \tag{1}$$

$$\frac{\partial hv}{\partial t} + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial Z}{\partial y}$$

where x and y are the spatial coordinates, t is the time, h is the height of the flow above the bottom Z, g is acceleration of gravity and u and v are the velocity components in the x and y directions. In the standard shallow-water formulation the bottom is a function of space only, that is $Z = Z(x, y)$. To include the effect of sediment transport an additional

equation which describes the time evolution of the bed level is required. In this work we use the Exner equation, a well-known and a common choice for this kind of works:

$$(1-p)\,\frac{\partial Z}{\partial t}+\frac{\partial Q_x}{\partial x}+\frac{\partial Q_y}{\partial y}=0 \tag{2}$$

where p is the (constant) sediment porosity and Q_x and Q_y are the bed-load sediment transport fluxes in the x and y directions. Many different formulations are possible for the definition of these fluxes [1]: for the sake of simplicity in this work we restrict our attention to the Grass model [2]:

$$Q_x = Au\left(u^2+v^2\right)^{\frac{m-1}{2}}, \quad Q_y = Av\left(u^2+v^2\right)^{\frac{m-1}{2}} \tag{3}$$

where A and $1 \leq m \leq 4$ are experimental constants depending on the particular problem under consideration.

3 Numerical Method

The numerical method proposed in this work to discretize in space the system of equations (1)-(3) is a finite-volume approach, applicable to unstructured grids. For the sake of simplicity we rewrite the system (1)-(3) in the form:

$$\frac{\partial \mathbf{W}}{\partial t}+\frac{\partial \mathbf{F}(\mathbf{W})}{\partial x}+\frac{\mathbf{G}(\mathbf{W})}{\partial y}=\mathbf{S}(\mathbf{W}) \tag{4}$$

where $\mathbf{W}$, $\mathbf{F}(\mathbf{W})$, $\mathbf{G}(\mathbf{W})$ and $\mathbf{S}(\mathbf{W})$ are defined as follows:

$$\mathbf{W}=\begin{pmatrix} h \\ hu \\ hv \\ Z \end{pmatrix},\ \mathbf{S}(\mathbf{W})=\begin{pmatrix} 0 \\ -gh\dfrac{\partial Z}{\partial x} \\ -gh\dfrac{\partial Z}{\partial y} \\ 0 \end{pmatrix}, \tag{5}$$

$$\mathbf{F}(\mathbf{W})=\begin{pmatrix} hu \\ hu^2+\dfrac{1}{2}gh^2 \\ huv \\ \dfrac{Q_x}{1-p} \end{pmatrix},\ \mathbf{G}(\mathbf{W})=\begin{pmatrix} hv \\ huv \\ hv^2+\dfrac{1}{2}gh^2 \\ \dfrac{Q_y}{1-p} \end{pmatrix}, \tag{6}$$

Starting from (4), a first-order general finite-volume discretization is obtained:

$$\frac{\mathbf{W}_i^{n+1}-\mathbf{W}_i^n}{\Delta t}=-\frac{1}{|V_i|}\sum_{j\in N(i)}\int_{\Gamma_{ij}}\mathcal{F}(\mathbf{W}^{n^*},\mathbf{n})d\sigma+\frac{1}{|V_i|}\int_{V_i}\mathbf{S}(\mathbf{W}^{n^*})dV \tag{7}$$

where $N(i)$ is the set of neighbouring cells of the i^{th} cell, $\mathbf{W}_i^n$ is an average value of the solution $\mathbf{W}$ in the i^{th} cell at time t_n, $|V_i|$ is the area of the cell, Γ_{ij} is the interface between cell i and j and $\mathbf{n} = (n_x, n_y)$ denotes the unit vector normal to Γ_{ij}. The finite volume discretization (7) is complete once a definition for the flux function $\mathcal{F}$ at the cell interfaces is given. Finally the time index n^* can be chosen to take the value n or $n+1$: the first choice corresponds to an explicit time scheme, the second one to an implicit scheme.

3.1 Explicit time advancing

In this work we consider the SRNH (Non-Homogeneous Riemann Solver) scheme introduced in [3]. Here we give only a brief summary of the main characteristics of the scheme, for a complete description of the numerical method we refer to [3, 4]. The scheme is composed by a predictor and a corrector stage: in the predictor stage an averaged state $\mathbf{U}_{ij}^n$ is computed, then this predicted state is used in the corrector stage to update the solution.

The predictor stage is based on primitive variables projected on the normal and tangential directions with respect to the cell interface (η and τ respectively). Hence, by defining

$$u_\eta = u\eta_x + v\eta_y, \quad u_\tau = -u\eta_y + v\eta_x \tag{8}$$

and assuming no space variation in the τ direction, it is possible to reformulate the system (4) as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_\eta(\mathbf{U})\frac{\partial \mathbf{U}}{\partial \eta} = 0 \tag{9}$$

where

$$\mathbf{U} = \begin{pmatrix} h \\ u_\eta \\ u_\tau \\ Z \end{pmatrix}, \ \mathbf{A}_\eta(\mathbf{U}) = \begin{pmatrix} u_\eta & h & 0 & 0 \\ g & u_\eta & 0 & g \\ 0 & 0 & u_\eta & 0 \\ 0 & A\xi(3u_\eta^2 + u_\tau^2) & 2A\xi u_\eta u_\tau & 0 \end{pmatrix} \tag{10}$$

To complete the description of the SNRH scheme, it is necessary to introduce the sign matrix and the Roe averaged state. The sign matrix is defined as:

$$\text{sgn}\left[\mathbf{A}_\eta(\overline{\mathbf{U}})\right] = \mathcal{R}(\overline{\mathbf{U}})\text{sgn}\left[\Lambda(\overline{\mathbf{U}})\right]\mathcal{R}^{-1}(\overline{\mathbf{U}}) \tag{11}$$

where $\Lambda(\overline{\mathbf{U}})$ is the diagonal matrix of eigenvalues of $\mathbf{A}_\eta(\overline{\mathbf{U}})$, and $\mathcal{R}(\overline{\mathbf{U}})$ is the corresponding right-eigenvector matrix. $\overline{\mathbf{U}}_{ij}$ is the Roe averaged state, defined as:

$$\overline{\mathbf{U}}_{ij} = \begin{pmatrix} \dfrac{h_i + h_j}{2} \\ \dfrac{u_{\eta,i}\sqrt{h_i} + u_{\eta,j}\sqrt{h_j}}{\sqrt{h_i} + \sqrt{h_j}} \\ \dfrac{u_{\tau,i}\sqrt{h_i} + u_{\tau,j}\sqrt{h_j}}{\sqrt{h_i} + \sqrt{h_j}} \\ \dfrac{Z_i + Z_j}{2} \end{pmatrix} \tag{12}$$

The predictor and corrector stages of the explicit SRNH scheme are then formulated as follows:

$$\begin{aligned} \mathbf{U}_{ij}^n &= \frac{1}{2}\left(\mathbf{U}_i^n + \mathbf{U}_j^n\right) - \frac{1}{2}\mathrm{sgn}\left[\mathbf{A}_\eta(\overline{\mathbf{U}}_{ij})\right]\left(\mathbf{U}_j^n - \mathbf{U}_i^n\right) \\ \frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta t} &= -\frac{1}{|V_i|}\sum_{j\in N(i)} \mathcal{F}(\mathbf{W}_{ij}^n, \mathbf{n}_{ij})|\Gamma_{ij}| + \mathbf{S}_i^n \end{aligned} \tag{13}$$

where $\mathbf{S}_i^n$ is the discretization of the source term. The two equations (13) are coupled because $\mathbf{W}_{ij}^n$ is obtained from $\mathbf{U}_{ij}^n$ by inverting Eq. (8). In order to satisfy the C-property [5] the approximation of the source term is defined as follows:

$$\mathbf{S}_i^n = \begin{pmatrix} 0 \\ -g\overline{h}_{x,i}^n \displaystyle\sum_{j\in N(i)} Z_{ij}^n n_{x,ij}|\Gamma_{ij}| \\ -g\overline{h}_{y,i}^n \displaystyle\sum_{j\in N(i)} Z_{ij}^n n_{y,ij}|\Gamma_{ij}| \\ 0 \\ 0 \end{pmatrix} \tag{14}$$

with

$$\overline{h}_{x,i}^n = \frac{1}{2}\frac{\displaystyle\sum_{j\in N(i)} \left(h_{ij}^n\right)^2 n_{x,ij}|\Gamma_{ij}|}{\displaystyle\sum_{j\in N(i)} h_{ij}^n n_{x,ij}|\Gamma_{ij}|}, \quad \overline{h}_{y,i}^n = \frac{1}{2}\frac{\displaystyle\sum_{j\in N(i)} \left(h_{ij}^n\right)^2 n_{y,ij}|\Gamma_{ij}|}{\displaystyle\sum_{j\in N(i)} h_{ij}^n n_{y,ij}|\Gamma_{ij}|} \tag{15}$$

For additional details we refer the interested reader to [8, 9].

3.2 Implicit time advancing

To switch from an explicit scheme to an implicit one it is sufficient, from a theoretical point of view, to compute the quantities $\mathcal{F}(\mathbf{W}_{ij}^{n+1}, \mathbf{n}_{ij})$ and $\mathbf{S}_i^{n+1}$ instead of $\mathcal{F}(\mathbf{W}_{ij}^{n}, \mathbf{n}_{ij})$ and $\mathbf{S}_i^{n}$. However, from a practical point of view this would require the solution of a large non-linear system of equations at each time step. The computational cost for this operation is in general not affordable in practical applications and generally greatly overcomes any advantage that an implicit scheme could have with respect to its explicit counterpart. A common technique to overcome this difficulty is to linearize the numerical scheme, that is to find an approximation of $\mathcal{F}_{ij}^{n+1} = \mathcal{F}(\mathbf{W}_{ij}^{n+1}, \mathbf{n}_{ij})$ and $\mathbf{S}_i^{n+1}$ in the form:

$$\Delta^n \mathcal{F}_{ij} \simeq D_{1,ij}\Delta^n \mathbf{W}_i + D_{2,ij}\Delta^n \mathbf{W}_j, \quad \Delta^n \mathbf{S}_i \simeq \sum_{j \epsilon \bar{N}(i)} D_{3,ij}\Delta^n \mathbf{W}_j \tag{16}$$

where $\Delta^n(\cdot) = (\cdot)^{n+1} - (\cdot)^n$ and $\bar{N}(i) = N(i) \cup \{i\}$. Using this approximation, the following linear system must be solved at each time step:

$$\begin{aligned} \frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta t} + \frac{1}{|V_i|} \sum_{j \in N(i)} (D_{1,ij}\Delta^n \mathbf{W}_i + D_{2,ij}\Delta^n \mathbf{W}_j) - \sum_{j \epsilon \bar{N}(i)} D_{3,ij}\Delta^n \mathbf{W}_j \\ = -\frac{1}{|V_i|} \sum_{j \in N(i)} \mathcal{F}(\mathbf{W}_{ij}^n, \mathbf{n}_{ij}) |\Gamma_{ij}| + \mathbf{S}_i^n \end{aligned} \tag{17}$$

The implicit linearized scheme is completely defined once a suitable definition for the matrices $D_{1,ij}, D_{2,ij}, D_{3,ij}$ is given. If the flux function and the source term are differentiable, a common choice is to use the Jacobian matrices, hence:

$$\begin{cases} D_{1,ij} & \simeq & \left. \dfrac{\partial \mathcal{F}(\mathbf{W}_{ij}, \mathbf{n}_{ij})}{\partial \mathbf{W}_i} \right|_n \\ D_{2,ij} & \simeq & \left. \dfrac{\partial \mathcal{F}(\mathbf{W}_{ij}, \mathbf{n}_{ij})}{\partial \mathbf{W}_j} \right|_n \\ D_{3,ij} & \simeq & \left. \dfrac{\partial \mathbf{S}_i}{\partial \mathbf{W_j}} \right|_n \end{cases} \tag{18}$$

Nevertheless, it is not always possible nor convenient to exactly compute the Jacobian matrices. In fact, it is not unusual to have some lack of differentiability of the numerical flux functions. Furthermore the explicit scheme (13) is composed by a predictor and a corrector stage and this significantly increases the difficulty in using linearization (18). This problem has been solved herein through the use of the automatic differentiation software Tapenade [6]. The operational principle of an automatic differentiation software is as

follows: given the source code of a routine which computes the function $y = F(x)$, the automatic differentiation software generates a new source code which compute the analytical derivative of the original program. In practice each time the original program performs some operation, the differentiated program performs additional operations dealing with the differential values. For example, if the original program, at some time executes the following instruction on variables a, b, c:

$$a = b \cdot c \tag{19}$$

then the differentiated program computes also the differentials da, db, dc of these variable [6]:

$$da = db \cdot c + b \cdot dc \tag{20}$$

Through an automatic differentiation software it is possible to quickly implement an implicit linearized scheme of the form (17), once a routine which computes the explicit flux function is available.

3.3 Extension to second order accuracy

The extension to second-order accuracy in space can be achieved by using a classical MUSCL technique [10], in which the flux function is computed by using the extrapolated variable values at the cell interface:

$$\mathbf{W}_{ij}^n = \mathbf{W}_i^n + \frac{1}{2}\nabla\mathbf{W}_i \cdot \mathbf{d}_{ij}, \quad \mathbf{W}_{ji}^n = \mathbf{W}_j^n - \frac{1}{2}\nabla\mathbf{W}_j \cdot \mathbf{d}_{ij} \tag{21}$$

where $\mathbf{d}_{ij}$ is the vector joining the i^{th} node with the j^{th} one and $\nabla\mathbf{W}_i$ is the cell gradient evaluated as in [11]. Note that in the computation of $\nabla\mathbf{W}$ we incorporate the Minmod slope limiter in order to obtain a TVD scheme.

For the explicit scheme, second-order accuracy in time is achieved through a two-step Runge-Kutta scheme for time advancing.

Finally, for the implicit case, it is possible to obtain a space and time second-order accurate formulation by considering the MUSCL technique for space as previously defined and a second-order backward differentiation formula in time:

$$|V_i|\frac{3\mathbf{W}_i^{n+1} - 4\mathbf{W}_i^n + \mathbf{W}_i^{n-1}}{2\Delta t} + \sum_{j\in N(i)} \Delta^n[\mathcal{F}_2]_{ij} - \Delta^n[\mathbf{S}_2]_i = -\sum_{j\in N(i)} [\mathcal{F}_2]_{ij}^n + [\mathbf{S}_2]_i^n \tag{22}$$

where $\mathcal{F}_2$ and $\mathbf{S}_2$ are the second-order accurate numerical flux and source terms computed as previously described. Similarly to the 1st-order case, a linearization of $\Delta^n\mathcal{F}_2$ and $\mathbf{S}_2$ must be carried out in order to avoid the solution of a nonlinear system at each time step. However, the linearization for the second-order accurate fluxes and the solution of

the resulting linear system implies significant computational costs and memory requirements. Thus, a defect-correction technique [7] is used here, which consists in iteratively solving simpler problems obtained, just considering the same linearization as used for the 1st-order scheme. The defect-correction iterations write as:

$$\begin{cases} \mathcal{W}^0 = \mathbf{W}^n \\ \\ \mathcal{B}_i^{i,s}\Delta^s\mathcal{W}_i + \displaystyle\sum_{j\in N(i)} \mathcal{B}_j^{i,s}\Delta^s\mathcal{W}_j = \mathcal{C}_i^s \quad s = 0, \cdots, r-1 \\ \\ \mathbf{W}^{n+1} = \mathcal{W}^r \end{cases} \tag{23}$$

in which:

$$\begin{cases} \mathcal{B}_i^{i,s} = \dfrac{3|V_i|}{2\Delta t}\, I + \displaystyle\sum_{j\in N(i)} D_{1,ij}(\mathcal{W}_i^s, \mathcal{W}_j^s) - D_{3,ii}(\mathbf{S}_i^s) \\ \\ \mathcal{B}_j^{i,s} = D_{2,ij}(\mathcal{W}_i^s, \mathcal{W}_j^s) - D_{3,ij}(\mathbf{S}_i^s) \\ \\ \mathcal{C}_i^s = -\left(\dfrac{|V_i|}{2\Delta t}(3\mathcal{W}_i^s - 4\mathbf{W}_i^n + \mathbf{W}_i^{n-1}) + \displaystyle\sum_{j\in N(i)} [\mathcal{F}_2]_{ij}^s - [\mathbf{S}_2]_i^s \right) \end{cases} \tag{24}$$

$D_{1,ij}, D_{2,ij}, D_{3,ij}$ being the generic matrices of the approximation (16); r is typically chosen equal to 2. Indeed, it can be shown [7, 12] that only one defect-correction iteration is theoretically needed to reach a second-order accuracy while few additional iterations (one or two) can improve the robustness.

4 One-dimensional numerical experiments

The first numerical experiment is the one dimensional test-case proposed in [4]. It is a sediment transport problem in a channel of length $l = 1000m$. The initial bottom topography is given by a hump shape function, that is:

$$Z(0,x) = \begin{cases} \sin^2\left(\dfrac{(x-300)\pi}{200}\right) & \text{if } 300 \le x \le 500 \\ \\ 0 & \text{elsewhere} \end{cases} \tag{25}$$

To generate the initial condition for the flow height and velocity field, we first simulated a problem without bed evolution. The initial condition for this standard shallow-water problem is:

$$h(0,x) = 10 - Z(0,x), \quad u(0,x) = \frac{10}{h(0,x)} \tag{26}$$

in which all the variables are in SI units. Starting from this condition the simulation has been run until a steady state has been reached. The obtained field has then been used as the initial condition for all the simulations with sediment transport. We compare the results between first and second order schemes, both explicit and implicit, in terms of accuracy and computational costs. Three sets of simulations have been carried out, characterized by $A = 0.001, 0.1, 1$ respectively, A being a free parameter in the Grass model (3). The first value corresponds to a slow interaction between the flow and the bed-load, the last to a fast interaction, and the second value to an intermediate case. In all cases the computational domain is discretized in 100 uniform cells. The simulations have been carried out for 238000, 2380 and 238 secs for $A = 0.001$, $A = 0.1$ and $A = 1$ respectively. All the results and CPU times shown in the following are at the final instant of each simulation. All the 1D simulations have been carried out on a laptop having a AMD 3000+ processor with 2Gb RAM and only one defect-correction iteration has been used.

4.1 Slow interaction between bed-load and water flow

Figures 1a and 1b show a comparison of the results obtained by means of the explicit scheme at $CFL = 0.8$ with those of the implicit scheme $CFL = 100$, both for 1st and 2nd-order accuracy. There is practically no difference between the solutions obtained with the implicit and explicit schemes, while the results obtained at first-order of accuracy significantly differ from the second-order ones. As for the efficiency, the implicit scheme

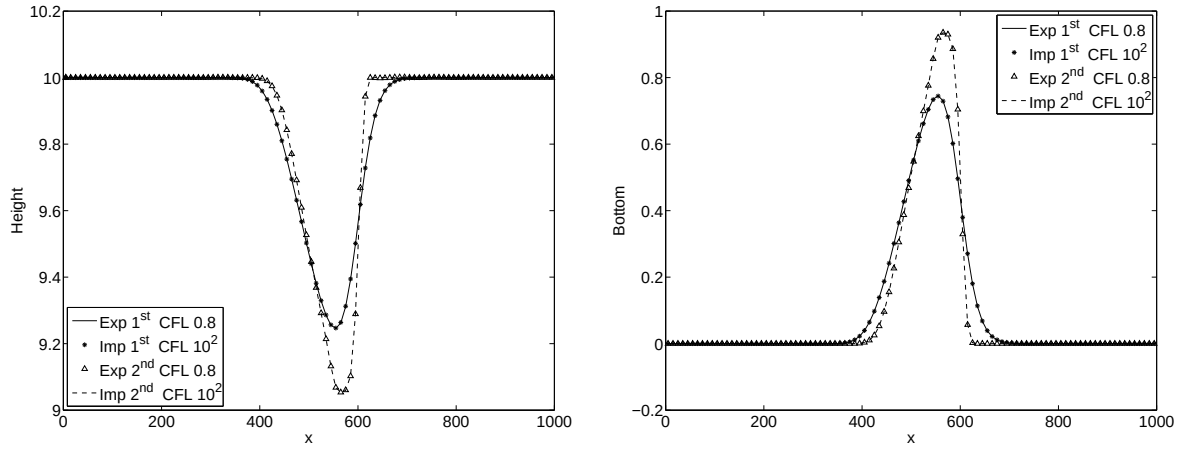

Figure 1: Comparison of the results of the explicit and implicit schemes, for first and second-order of accuracy: A $=10^{-3}$; (left) height, (right) bottom.

seems to be unconditionally stable: the CFL has been increased up to 10^5 while obtaining stable solutions. However, the accuracy of the results obviously decreases if the time step is too large. The results obtained with the implicit scheme at different values of the CFL are compared in Fig 2. In this case of slow interaction between bed-load and water

flow, the quality of the results for the implicit scheme is not significantly deteriorated up to a CFL number of 1000. As for computational costs, Tab. 1 shows that already at $CFL = 100$ the gain in CPU time obtained with the implicit scheme is large, both at first and second-order of accuracy. Note also that in this case, in which only one defect-correction iteration is made, the CPU gain obtained with the implicit scheme is significantly larger for second-order accuracy.

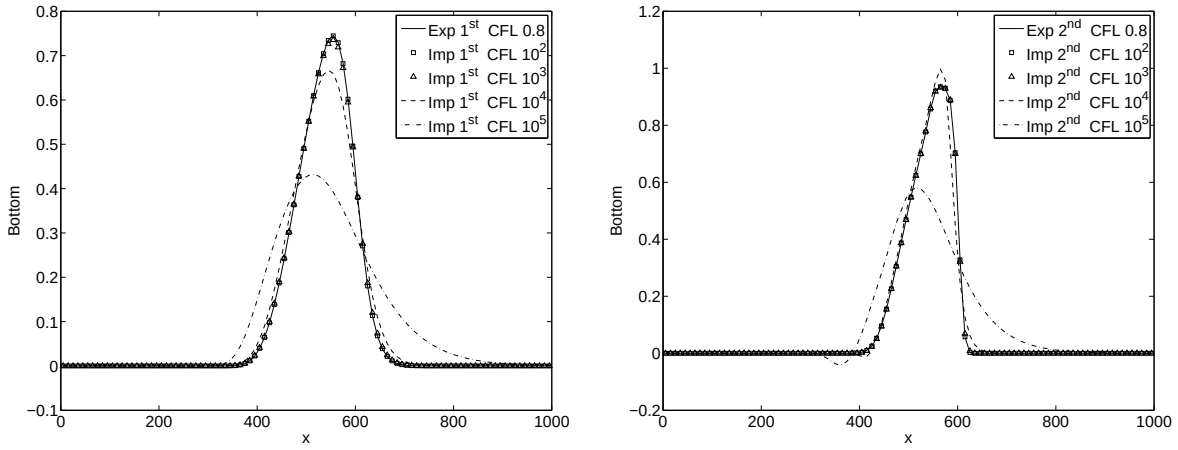

Figure 2: Bottom height obtained with the explicit scheme at CFL 0.8 and with the implicit one at different values of CFL, A $=10^{-3}$, first-order (left) and second-order (right) of accuracy.

Explicit, CFL 0.8		Implicit, CFL 100	
1^{st} order	2^{nd} order	1^{st} order	2^{nd} order
$43s$	$63.7s$	$2.2s$	$2.4s$

Table 1: CPU time required (seconds), A $=10^{-3}$.

4.2 Intermediate speed of interaction between bed-load and water flow

Figures 3a and 3b show a comparison of the results obtained by means of the explicit scheme at $CFL = 0.8$ with those of the implicit scheme at $CFL = 10$, both for 1st and 2nd-order accuracy. As for the previous case there is practically no difference between the solutions obtained with the implicit and explicit schemes, while the results obtained at first-order of accuracy significantly differ from the second-order ones. Also in this case, the implicit scheme seems to be unconditionally stable: the CFL has been increased up to 10^4 (only 3 iterations to complete the simulation). The results obtained with the

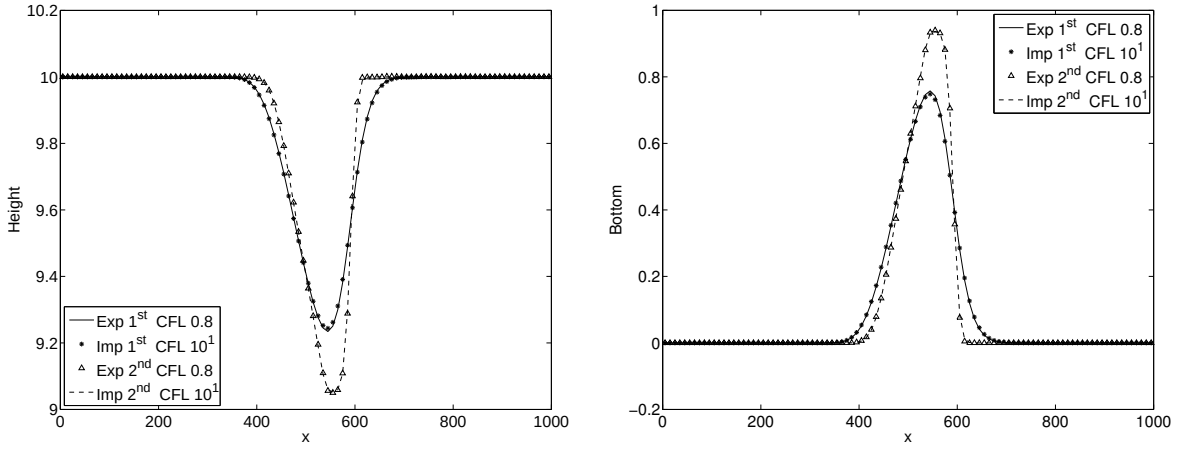

Figure 3: Comparison of the results of the explicit and implicit schemes, for first and second-order of accuracy: A $=10^{-1}$; (left) height, (right) bottom.

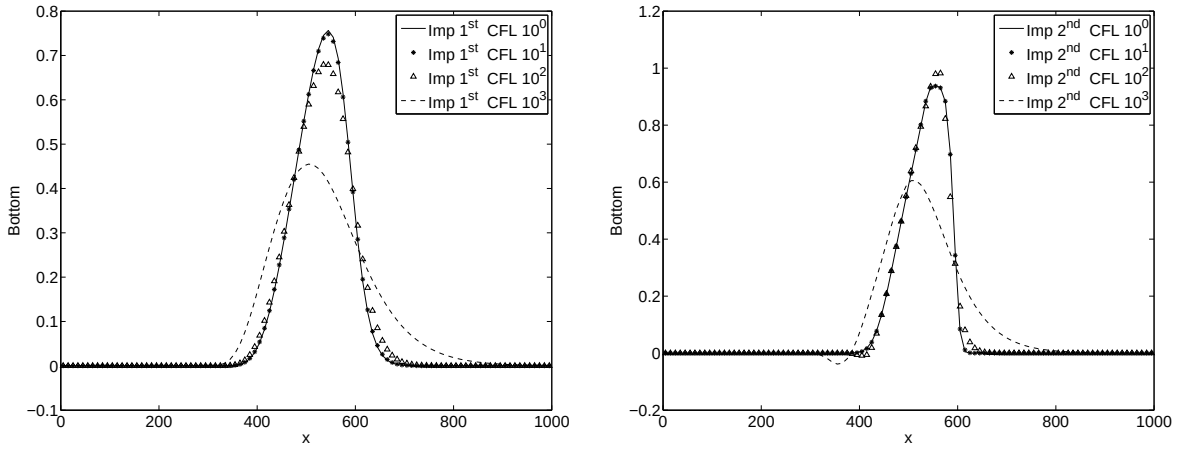

Figure 4: Bottom height obtained with the explicit scheme at CFL 0.8 and with the implicit one at different values of CFL, A $=10^{-1}$, first-order (left) and second-order (right) of accuracy.

implicit scheme at different values of the CFL are compared in Fig 4. In this case of intermediate interaction between bed-load and water flow, the quality of the results for the implicit scheme is not affected up to a CFL number of 100. As for computational costs, Tab. 2 shows that at $CFL = 100$ there is a gain in CPU time obtained with the implicit scheme, both at first and second-order of accuracy even if it is not as large as the one in the previous case.

4.3 Fast interaction bed-load/water flow

Figures 5a and 5b show a comparison of the results obtained by means of the explicit scheme at $CFL = 0.8$ with those of the implicit scheme $CFL = 1$, both for 1st and

Explicit, CFL 0.8		Implicit, CFL 100	
1^{st} order	2^{nd} order	1^{st} order	2^{nd} order
$1.3s$	$1.7s$	$0.6s$	$0.7s$

Table 2: CPU time (seconds) for the Shallow Water equations with A $=10^{-1}$.

2nd-order accuracy. As for the previous case there is no difference between the solutions obtained with the implicit and explicit schemes.

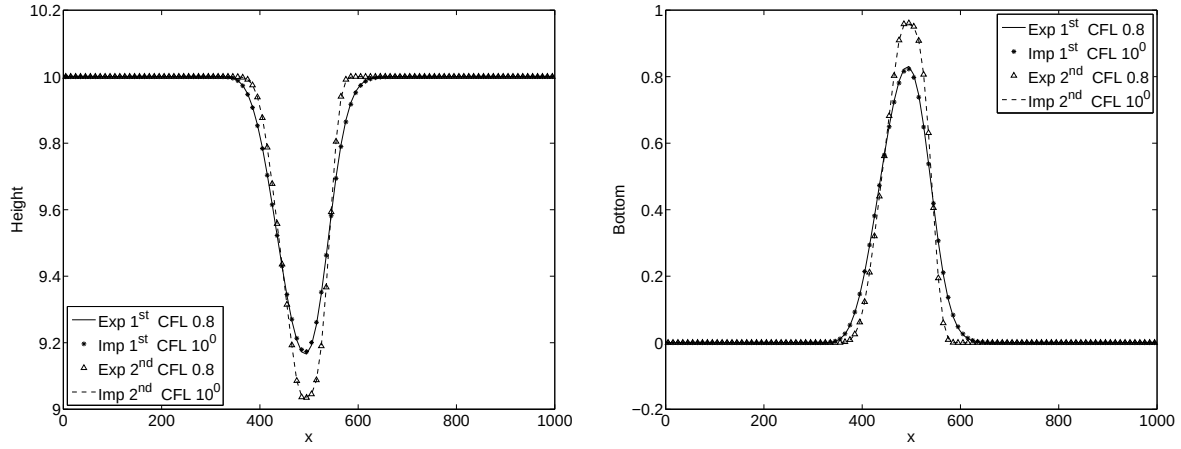

Figure 5: Comparison of the results of the explicit and implicit schemes, for first and second-order of accuracy: A =1; (a) height, (b) bottom.

In this case of fast interaction between bed-load and water flow, the quality of the results for the implicit scheme imposes a maximum CFL number equal to 1, although the implicit scheme seems again to be unconditionally stable. As a consequence in this test case the computational cost of the implicit scheme is larger than for the explicit one, both at first and second order of accuracy, as it is shown in Tab. 3. Summarizing, the

Explicit, CFL 0.8		Implicit, CFL 1	
1^{st} order	2^{nd} order	1^{st} order	2^{nd} order
$0.13s$	$0.2s$	$0.46s$	$0.52s$

Table 3: CPU Time (seconds), A =1.

implicit scheme seems to be unconditionally stable in all the considered cases and the CFL limitation to avoid loss of accuracy is roughly inversely proportional to A.

5 2D Results

The 2D test case considered herein is the one proposed in [4]. The initial bottom topography is defined as follows:

$$Z(0,x) = \begin{cases} \sin^2\left(\frac{(x-300)\pi}{200}\right)\sin^2\left(\frac{(y-400)\pi}{200}\right) & \text{if } (x,y) \in Q_h \\ 0 & \text{elsewhere} \end{cases} \tag{27}$$

where $Q_h = [300, 500] \times [400, 600]$. The computational domain is a rectangle of length $6000 \times 1400\ m^2$. Dirichlet boundary conditions are imposed for velocity at the inlet, while at the outlet characteristic based conditions are used. Finally, free-slip is imposed on the lateral boundaries. As in the 1D simulations, in order to compute a suitable initial condition for the sediment transport problem, we first solved the standard shallow-water problem with constant bottom and the following initial condition:

$$h(0,x,y) = 10 - Z(0,x,y) \quad u(0,x,y) = \frac{10}{h(0,x,y)} \tag{28}$$

Once a steady condition is reached, this field is used as initial condition for the sediment transport problem. The used grids are divided into two main regions: a square region of 1000×1000 meters where the mesh is refined in order to capture the main characteristic of the flow field, and an external region in which a coarse mesh is used, in order to limit the influence of boundary conditions. The main characteristics of the grids used in our simulations are reported in table 4, while a snapshot of the coarser grid $GR1$ is shown in figure 6.

Grid	Nodes	Elements	l_r
GR1	18265	36112	10
GR2	29191	57924	5

Table 4: Main characteristics of the grids used in the simulations, l_r is the dimension of the elements in the refined region.

The 2D numerical experiments correspond to a slow bed-load/flow interaction ($A = 10^{-3}$). The initial conditions are shown in figure 7 on grid GR1. It has been found that for this value of the parameter A the implicit scheme seems to be stable for CFL numbers up to 100, while the explicit simulations reach a maximum CFL number of 0.8. Figure 8

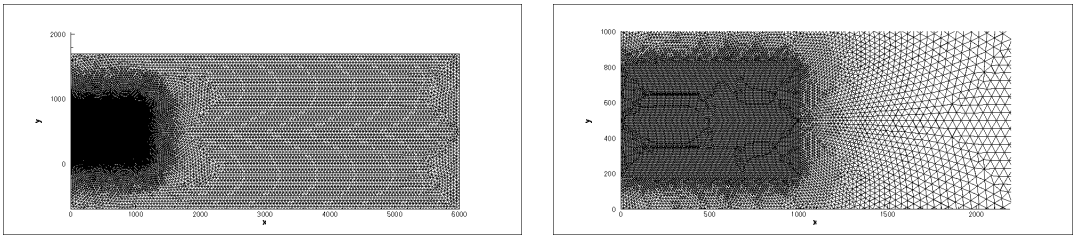

Figure 6: Grid GR1: global view (left) and zoom of the refined region(right)

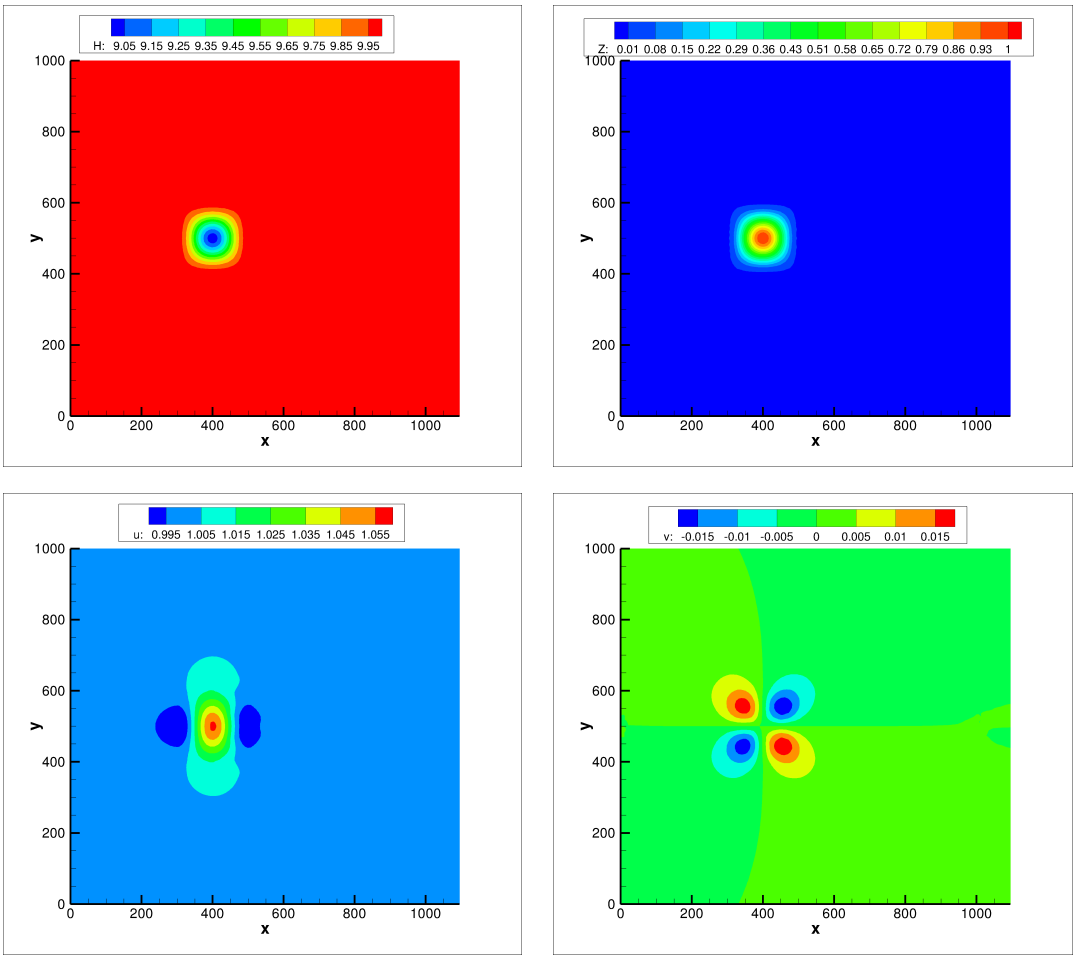

Figure 7: Initial conditions on grid GR1, from left to right and from top to bottom: height of the flow, bottom, and the two components of the velocity

shows a comparison of the results obtained by the implicit and the explicit schemes after a time of 20 hours for the two different grids. All the 2D results shown herein are 2nd-order accurate. As in the 1D case there is practically no difference between the results obtained

on the same grid by the implicit and explicit schemes. The solutions are also in good agreement with the results in [4].

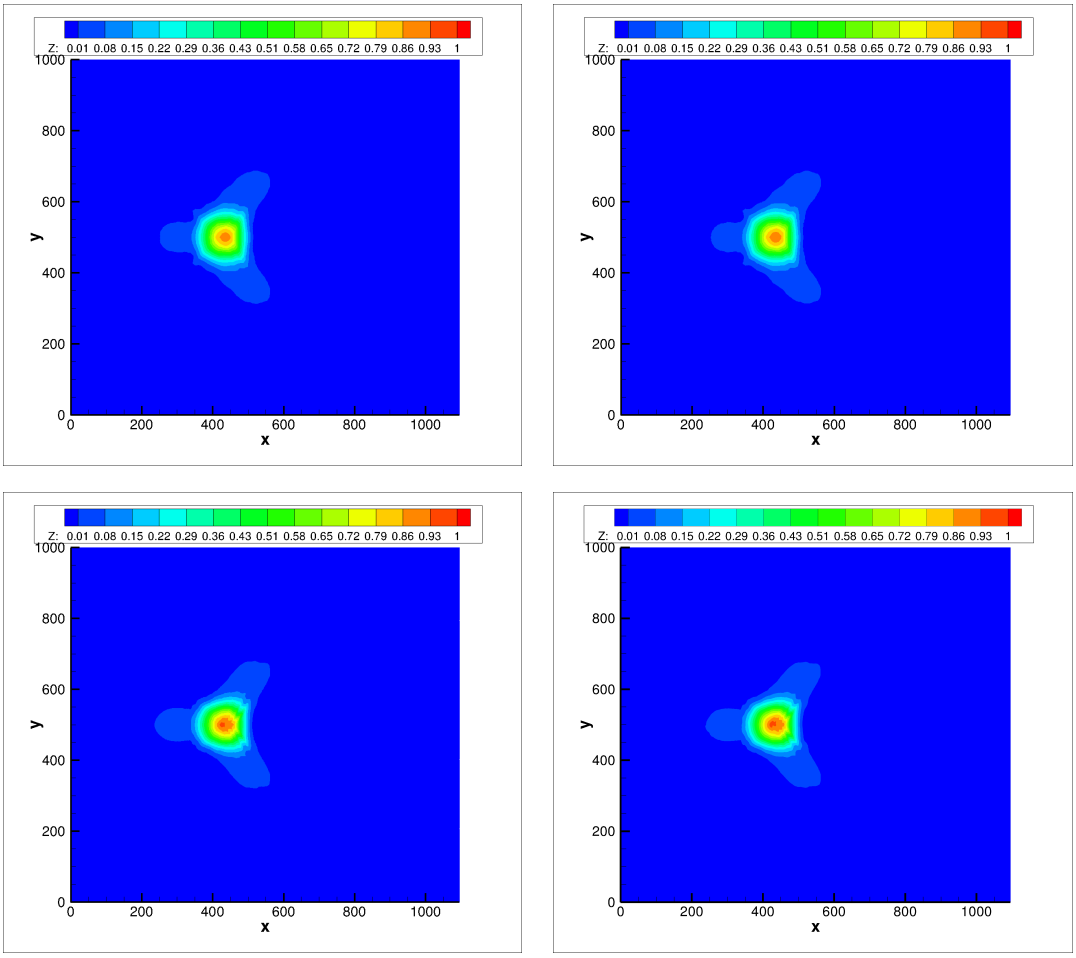

Figure 8: Comparison of the results after 20 hours of simulation between the explicit (left) and implicit (right) schemes: top grid GR1, bottom GR2

The CPU time required by the simulations is shown in Tab. 5. The simulations have been carried out on a PC having an Intel Xeon 3.06 GHz processor with 2GB of RAM. The CPU times in Tab. 5 only give a relative comparison between the schemes, while their values are only approximate since the computer was not completely dedicated to the task. Note that all the 2D implicit simulations have been run with two iterations of defect correction. The CPU time required by the explicit scheme in this case is clearly larger than the corresponding time required by the implicit scheme. Note also that with only one Defect correction iteration the implicit scheme does not appear to be stable for the CFL value of 100. We are currently investigating whether a further increment of the number of Defect Correction iterations can further improve the stability range of the

	Explicit	Implicit
GR1	$\simeq$ 24 h	$\simeq$ 4 h
GR2	$\simeq$ 50 h	$\simeq$ 8 h

Table 5: CPU time required by the numerical simulations

implicit scheme. Also the influence of different methods of solution for the linear system associated to the implicit scheme could be of interest.

6 Concluding remarks

The numerical simulation of sediment transport problems has been considered. The problem has been modeled through the shallow-water equations coupled with the Exner equation to describe the time evolution of the bed level. The Grass model has been used for the sediment transport. The governing equations have been discretized by using a finite-volume method together with the SRHN predictor-corrector scheme. Starting from the explicit version, a linearized implicit scheme has been built, in which the flux Jacobians are computed through automatic differentiation. This allowed the complexity of the analytical differentiation of the predictor-corrector scheme to be avoided. Second-order accuracy in space and time has been obtained through MUSCL reconstruction together with a defect-correction approach.

The focus of the present paper was on the comparison between implicit and explicit schemes, in terms of accuracy and computational requirements. 1D and 2D test cases have been considered. The 1D cases are characterized by different rates of interaction between the bed and the water flow, while the 2D numerical experiments have been carried out for a slow interaction, which a-priori is the most favorable case for implicit schemes. In the 1D simulations, the implicit method was found to run with a virtually unlimited CFL number without stability problems. However, to avoid loss of accuracy, the CFL number of the implicit scheme must be reduced to a value roughly inversely proportional to the velocity of the interaction between the flow and the bed-load. Despite this limitation, the implicit code has been found to be computationally more efficient than the explicit one for slow and intermediate rates of the interaction between the bed and the flow. The 2D tests seem to confirm these results, at least for a slow bedload/water-flow interaction, since the implicit code has been found to be significantly cheaper than its explicit counterpart and the results obtained on the same grid are practically the same. However, to better assess the capabilities of the implicit scheme in 2D, further investigations are ongoing.

REFERENCES

[1] M.J. Castro Díaz, E.D. Fernández-Nieto, and A.M. Ferreiro. Sediment transport models in shallow water equations and numerical approach by high order finite volume methods. *Computers & Fluids*, 37(3):299–316, 2008.

[2] A.J. Grass. Sediments transport by waves and currents. Technical report, SERC London Cent. Mar. Technol., Report No. FL29, 1981.

[3] S. Sahmim, F. Benkhaldoun, and F. Alcrudo. A sign matrix based scheme for non-homogeneous PDE's with an analysis of the convergence stagnation phenomenon. *Journal of Computational Physics*, 226(2):1753–1783, 2007.

[4] F. Benkhaldoun, S. Sahmim, and M. Seaïd. A two-dimensional finite volume morphodynamic model on unstructured triangular grids. *International Journal for Numerical Methods in Fluids*, In press.

[5] A. Bermudez and M.E. Vazquez. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids*, 23(8):1049–1071, 1994.

[6] L. Hascoët and V. Pascual. *TAPENADE 2.1 User's Guide*. Technical Report n 300. INRIA, 2004.

[7] R. Martin and H. Guillard. A second order defect correction scheme for unsteady problems. *Computers & Fluids*, 25(1):9–27, 1996.

[8] F. Benkhaldoun, I. Elmahi, and M. Seaïd. Well-balanced finite volume schemes for pollutant transport by shallow water equations on unstructured meshes. *Journal of Computational Physics*, 226(1):180–203, 2007.

[9] F. Benkhaldoun, S. Sahmim, and M. Seaïd. Solution of the sediment transport equations using a finite volume method based on sign matrix. *SIAM Journal on Scientific Computing*, 31(4):2866–2889, 2009.

[10] B. van Leer. Towards the ultimate conservative difference scheme V: a second-order sequel to Godunov's method. *J. Comput. Phys.*, 32(1):101–136, 1979.

[11] S. Camarri, M.V. Salvetti, B. Koobus, and A. Dervieux. A low-diffusion MUSCL scheme for LES on unstructured grids. *Computers & Fluids*, 33:1101–1129, 2004.

[12] E. Sinibaldi, F. Beux, M. Bilanceri, and M.V. Salvetti. *A Second-Order Linearised Implicit Formulation for Hyperbolic Conservation Laws, with Application to Barotropic Flows*. ADIA 2008-4. University of Pisa, June 2008.