Comparison of Explicit and Implicit Time Advancing in the Simulation of a 2D Sediment Transport Problem

M. Bilanceri, F. Beux, I. Elmahi, H. Guillard, and M.V. Salvetti

Abstract The simulation of sediment transport, based on the shallow-water equations coupled with Grass model for the sediment transport equation is considered. The aim of the present paper is to investigate the behavior of implicit linearized schemes in this context. A finite-volume method is considered and second-order accuracy in space is obtained through MUSCL reconstruction. A second-order time accurate explicit version of the scheme is obtained through a two step Runge-Kutta method. Implicit linearized schemes of second-order of accuracy in time are derived thanks to a BDF method associated with a Defect Correction technique. The different time-advancing schemes are compared, using a 2D sediment transport problem, with different types of flow/bed interactions. The implicit one largely outperforms the explicit version for slow flow/bed interactions while in the case of fast flow/bed interactions, the CPU time of both time integration schemes are comparable. Thus, the implicit scheme turns out to be a good candidate to simulate flows with sediment transport in practical applications.

Keywords sediment transport, Grass model, linearized implicit time advancing, automatic differentiation
MSC2010: 65-06

M. Bilanceri and M.V. Salvetti
University of Pisa (Italy), e-mail: marco.bilanceri@gmail.com, mv.salvetti@ing.unipi.it

F. Beux
Alta S.p.A., Pisa (Italy), e-mail: f.beux@alta-space.com

I. Elmahi
EMCS, Ensa, Oujda, Complexe Universitaire, Oujda (Morocco), e-mail: ielmahi@ensa.ump.ma

H. Guillard
INRIA, Sophia Antipolis and Laboratoire Jean-Alexandre Dieudonné, University of Nice Sophia-Antipolis, Parc Valrose 06108 Nice Cedex, (France), e-mail: Herve.Guillard@sophia.inria.fr

J. Fořt et al. (eds.), *Finite Volumes for Complex Applications VI – Problems & Perspectives*, Springer Proceedings in Mathematics 4,
DOI 10.1007/978-3-642-20671-9_14,

1 Introduction

A huge amount of work has been done in the last decades to develop numerical methods for the simulation of sediment transport problems (see, e.g., the references in [1, 4]). In this context, the hydrodynamics part is usually modeled through the classical shallow-water equations coupled with an additional equation for the morphodynamical component. The Grass equation [6] is considered herein, which is one of the most popular and simple models. In this context, the treatment of the source terms and of the bed-load fluxes has received the largest attention while 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 a 2D sediment transport problem. We only consider flows over wet areas. The extension to cases in presence of dry areas will the object of further studies. If the interaction of the water flow with the mobile bed is slow, the characteristic time scales of the flow and of the sediment transport can be very different introducing time stiffness in the global problem. Thus, for these cases, it can be advantageous to use implicit schemes. On the other hand, since the considered problems are unsteady, attention must be paid for implicit schemes in the choice of the time step. 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. In order to overcome these difficulties, we use an automatic differentiation tool (Tapenade, [7]). Our starting point was the SRNH numerical scheme, specifically developed and validated for the numerical simulation of sediment transport problems [1]. An implicit version of this scheme 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 [10] is finally used to obtain second-order accuracy at limited computational costs. The implicit method is compared with the explicit one in a 2D benchmark.

2 Physical model and Numerical Method

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:

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x} + \frac{\mathbf{G}(\mathbf{W})}{\partial y} = \mathbf{S}(\mathbf{W}) \tag{1}$$

where x and y are the spatial coordinates, t is the time, and $\mathbf{W}$, $\mathbf{F}(\mathbf{W})$, $\mathbf{G}(\mathbf{W})$ and $\mathbf{S}(\mathbf{W})$ are defined as follows:

$$\begin{cases} \mathbf{W} &= \left(h, \quad hu, \quad hv, \quad Z \right)^T \\ \mathbf{F}(\mathbf{W}) &= \left(hu, \quad hu^2 + \frac{1}{2}gh^2 + ghZ, \quad huv, \quad \frac{1}{1-p}Q_x \right)^T \\ \mathbf{G}(\mathbf{W}) &= \left(hv, \quad hvu, \quad hv^2 + \frac{1}{2}gh^2 + ghZ, \quad \frac{1}{1-p}Q_y \right)^T \\ \mathbf{S}(\mathbf{W}) &= \left(0, \quad gZ\frac{\partial h}{\partial x}, \quad gZ\frac{\partial h}{\partial y}, \quad 0 \right)^T \end{cases} \tag{2}$$

In (2) 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. The first three equations of (1) are the standard 2D Shallow Water equations, recast as in [8] in order to avoid the singularity of the Jacobian of the flux function. The last one is the well-known Exner equation for the evolution of the bed level. We restrict our attention to the case in which the sediment transport porosity p is constant and the bed-load sediment transport fluxes Q_x and Q_y are defined by the Grass model:

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

where A and $1 \leq m \leq 4$ are experimental constants depending on the particular problem under consideration. The classical case $m = 3$ is considered here.

The numerical method proposed to discretize in space the system of equations (1)-(2) is a finite-volume approach, applicable to unstructured grids. Namely, it is the SRNH scheme introduced in [11]. A brief summary of the main characteristics of the scheme is given herein, for additional details we refer to [1, 11].

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, $\mathbf{n}$ and τ. Hence, by introducing the normal and tangential components of the velocity, $u_\mathbf{n}$ and u_τ, it is possible to reformulate the system (1) as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_\mathbf{n}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial \mathbf{n}} = 0 \tag{4}$$

$$\mathbf{U} = \begin{pmatrix} h \\ u_\mathbf{n} \\ u_\tau \\ Z \end{pmatrix}, \quad \mathbf{A}_\mathbf{n}(\mathbf{U}) = \begin{pmatrix} u_\mathbf{n} & h & 0 & 0 \\ g & u_\mathbf{n} & 0 & g \\ 0 & 0 & u_\mathbf{n} & 0 \\ 0 & A(1-p)^{-1}(3u_\mathbf{n}^2 + u_\tau^2) & 2A(1-p)^{-1}u_\mathbf{n}u_\tau & 0 \end{pmatrix} \tag{5}$$

Starting from (5) it is possible to introduce a Roe average state $\overline{\mathbf{U}}_{ij}$ and a sign matrix $\text{sgn}\left[\mathbf{A}_\mathbf{n}(\overline{\mathbf{U}})\right]$ defined as:

$$\overline{\mathbf{U}}_{ij} = \left(\frac{h_i + h_j}{2}, \frac{u_{\mathbf{n},i}\sqrt{h_i} + u_{\mathbf{n},j}\sqrt{h_j}}{\sqrt{h_i} + \sqrt{h_j}}, \frac{u_{\tau,i}\sqrt{h_i} + u_{\tau,j}\sqrt{h_j}}{\sqrt{h_i} + \sqrt{h_j}}, \frac{Z_i + Z_j}{2} \right)^T \tag{6}$$

$$\operatorname{sgn}\left[\mathbf{A}_{\mathbf{n}}(\overline{\mathbf{U}})\right] = \mathscr{R}(\overline{\mathbf{U}}) \Lambda_{\operatorname{sgn}}(\overline{\mathbf{U}}) \mathscr{R}^{-1}(\overline{\mathbf{U}}) \tag{7}$$

where the elements of the diagonal matrix $\Lambda_{\operatorname{sgn}}(\overline{\mathbf{U}})$ are the sign function of the eigenvalues of $\mathbf{A}_{\mathbf{n}}(\overline{\mathbf{U}})$ and $\mathscr{R}(\overline{\mathbf{U}})$ is the corresponding right-eigenvector matrix.

The explicit SRNH scheme is then formulated as follows:

$$\mathbf{U}_{ij}^n = \frac{1}{2}\left(\mathbf{U}_i^n + \mathbf{U}_j^n\right) - \frac{1}{2}\operatorname{sgn}\left[\mathbf{A}_{\mathbf{n}}(\overline{\mathbf{U}}_{ij})\right]\left(\mathbf{U}_j^n - \mathbf{U}_i^n\right) \tag{8}$$

$$\frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta^n t} = -\frac{1}{|V_i|} \sum_{j \in N(i)} \mathscr{F}(\mathbf{W}_{ij}^n, \mathbf{n}_{ij}) |\Gamma_{ij}| + \mathbf{S}_i^n \tag{9}$$

where $\mathbf{W}_{ij}^n$ is obtained from $\mathbf{U}_{ij}^n$, $N(i)$ is the set of neighboring cells of the i^{th} cell, $|V_i|$ is the area of the cell, Γ_{ij} is the interface between cell i and j, $\Delta^n t$ is the n^{th} time-step and $\mathscr{F}$ is the analytical flux function. $\mathbf{S}_i^n$ is the discretization of the source term which, in order to satisfy the C-property [2] is defined as follows:

$$\begin{cases} \overline{Z}_{x,i}^n = \dfrac{1}{2} \dfrac{\displaystyle\sum_{j \in N(i)} \left(Z_{ij}^n\right)^2 n_{x,ij} |\Gamma_{ij}|}{\displaystyle\sum_{j \in N(i)} Z_{ij}^n n_{x,ij} |\Gamma_{ij}|}, \quad \overline{Z}_{y,i}^n = \dfrac{1}{2} \dfrac{\displaystyle\sum_{j \in N(i)} \left(Z_{ij}^n\right)^2 n_{y,ij} |\Gamma_{ij}|}{\displaystyle\sum_{j \in N(i)} Z_{ij}^n n_{y,ij} |\Gamma_{ij}|} \\ \mathbf{S}_i^n = \left(0, \quad g\overline{Z}_{x,i}^n \displaystyle\sum_{j \in N(i)} h_{ij}^n n_{x,ij} |\Gamma_{ij}|, \quad g\overline{Z}_{y,i}^n \sum_{j \in N(i)} h_{ij}^n n_{y,ij} |\Gamma_{ij}|, \quad 0\right)^T \end{cases} \tag{10}$$

To switch from an explicit scheme to an implicit one it is sufficient, to compute the quantities $\mathscr{F}_{ij}^{n+1} = \mathscr{F}(\mathbf{W}_{ij}^{n+1}, \mathbf{n}_{ij})$ and $\mathbf{S}_i^{n+1}$ instead of $\mathscr{F}(\mathbf{W}_{ij}^n, \mathbf{n}_{ij})$ 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, i.e. to find an approximation of $\mathscr{F}_{ij}^{n+1}$ and $\mathbf{S}_i^{n+1}$ in the form:

$$\Delta^n \mathscr{F}_{ij} \simeq D_{1,ij} \Delta^n \mathbf{W}_i + D_{2,ij} \Delta^n \mathbf{W}_j, \quad \Delta^n \mathbf{S}_i \simeq \sum_{j \in \bar{N}(i)} D_{3,ij} \Delta^n \mathbf{W}_j \tag{11}$$

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:

$$\frac{\mathbf{W}_i^{n+1}-\mathbf{W}_i^n}{\Delta t}+\frac{1}{|V_i|}\sum_{j\in N(i)}|\Gamma_{ij}|\left(D_{1,ij}\,\Delta^n\mathbf{W}_i+D_{2,ij}\,\Delta^n\mathbf{W}_j\right)-\sum_{j\in\bar{N}(i)}D_{3,ij}\,\Delta^n\mathbf{W}_j$$
$$=-\frac{1}{|V_i|}\sum_{j\in N(i)}\mathscr{F}(\mathbf{W}_{ij}^n,\mathbf{n}_{ij})|\Gamma_{ij}|+\mathbf{S}_i^n \quad (12)$$

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. 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 (9) is composed by a predictor and a corrector stage and this significantly increases the difficulty in linearizing. This problem has been solved herein by computing through the automatic differentiation software Tapenade [7] the flux Jacobians, which are used to approximate $\mathscr{F}_{ij}^{n+1}$ and $\mathbf{S}_i^{n+1}$, as defined in Eq. (11). Given the source code of a routine which computes the explicit numerical fluxes, the differentiation software generates a new source code which computes the flux Jacobians, and, thus, the derivation and the implementation of their analytical expressions can be avoided.

The extension to second-order accuracy in space can be achieved by using a classical MUSCL technique [9], in which (8) is computed by using extrapolated values at the cell interfaces. The extrapolation is done here as in [3] associated with the Minmod slope limiter. For the explicit scheme, second-order accuracy in time is achieved through a two-step Runge-Kutta scheme. Considering 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. However, the linearization for the second-order accurate fluxes and source terms and the solution of the resulting linear system implies significant computational costs and memory requirements. Thus, a defect-correction technique [10] is used here, which consists in iteratively solving simpler problems obtained, just considering the same linearization as used for the first-order scheme. Thus defining $\mathscr{W}^0=\mathbf{W}^n$, the defect-correction iterations write as follows, the unknown being $\Delta^s\mathscr{W}_i$:

$$\frac{(1+2\tau)}{\Delta^n t\,(1+\tau)}\Delta^s\mathscr{W}_i+\frac{1}{|V_i|}\sum_{j\in N(i)}|\Gamma_{ij}|\left(D_{1,ij}\,\Delta^s\mathscr{W}_i+D_{2,ij}\,\Delta^s\mathscr{W}_j\right)-\sum_{j\in\bar{N}(i)}D_{3,ij}\,\Delta^s\mathscr{W}_j$$
$$=\frac{(1+2\tau)\mathscr{W}_i^s-(1+\tau)^2\mathbf{W}_i^n+\tau^2\mathbf{W}_i^{n-1}}{\Delta^n t\,(1+\tau)}-\frac{1}{|V_i|}\sum_{j\in N(i)}\mathscr{F}\left(\mathscr{W}_{ij}^s,\mathscr{W}_{ji}^s\right)|\Gamma_{ij}|+[\mathbf{S}_2]_i^s \quad (13)$$

for $s=0,\cdots,r-1$. In (13), $\tau=\frac{\Delta^n t}{\Delta^{n-1}t}$, $D_{1,ij}, D_{2,ij}, D_{3,ij}$ are the matrices of the approximation (11) and the update solution is $\mathbf{W}^{n+1}=\mathscr{W}^r$. It can be shown

[10] 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.

3 Numerical Experiments

The 2D test case considered herein is a well-known benchmark test, proposed in several papers (see, e.g. [1,5]). It is a sediment transport problem in a square domain Ω of dimensions $1000 \times 1000\ m^2$ with a non constant bottom relief. The initial bottom topography is defined as follows:

$$Z(0,x) = \sin^2\left(\frac{(x-300)\pi}{200}\right)\sin^2\left(\frac{(y-400)\pi}{200}\right) \text{ if } (x,y) \in Q_h, \ \ 0 \text{ elsewhere} \tag{14}$$

where $Q_h = [300, 500] \times [400, 600]$. Given $Z(0, x)$, the remaining initial conditions are $h(0, x, y) = 10 - Z(0, x, y)$, $u(0, x, y) = \frac{10}{h(0,x,y)}$ and $v(0, x, y) = 0$. Considering the boundaries, Dirichlet boundary conditions are imposed at the inlet, while at the outlet characteristic based conditions are used. Finally, free-slip is imposed on the lateral boundaries. The spatial discretization of the computational domain has been carried out by using two different grids: for the first grid GR1, the number of the nodes and the characteristic length of the elements are, respectively, $l_m = 20\ m$ and $N_c = 2901$. The second grid GR2 is characterized by $l_m = 10\ m$ and $N_c = 11425$.

Two different values of the parameter A are considered, namely a case with slow interaction between the flow and the bed, $A = 0.001$ and a fast one, $A = 1$. Due to the different time scales for the evolution of the bottom topography, different time intervals have been simulated for the considered cases: the total simulation time is 500 seconds for $A = 1$ and 360000 seconds for $A = 0.001$.

For the slow speed of interaction case, Figure 1a shows a comparison of the results obtained by means of the explicit version of the scheme at CFL $= 0.8$ with those of the implicit one at CFL $= 1000$ both for 1^{st} and 2^{nd}-order accuracy for grid GR2. For the definition of the CFL number we refer to [1]. There is practically no difference between the solutions obtained with the implicit and explicit version of the schemes, while the results obtained at 1^{st}-order of accuracy significantly differ from the 2^{nd}-order ones. Note that the results shown in Fig. 1a for the 2^{nd}-order implicit scheme are computed using only one DeC iteration. By increasing the number of DeC iterations it is possible to further increment, without loosing in accuracy, the CFL number of the 2^{nd}-order implicit scheme (see Fig. 1b). In particular, when 3 DeC iterations are considered it is possible to use a CFL number equal to 10^4 (see also Table 1). As shown in Fig. 1b, similar results can be obtained by considering the grid GR1 instead of the GR2. The profiles of $h + Z$ are shown if Fig. 1c. Slightly larger oscillations are observed for the second-order implicit scheme, but at the first order both schemes gave practically the same results. As

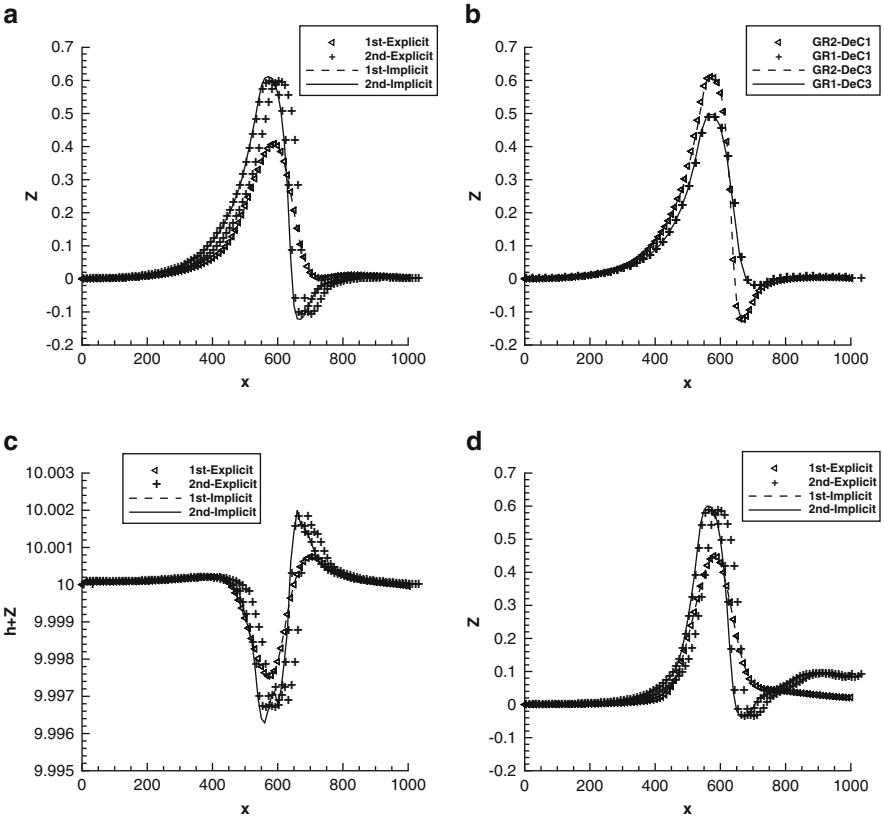

Fig. 1 Comparison of the results given the explicit and implicit schemes; profiles along the line $y = 500$ of: (a) Z for $A = 10^{-3}$ and GR2, (b) Z for implicit schemes and $A = 10^{-3}$, (c) $h + Z$ for $A = 10^{-3}$ GR2, (d) Z for $A = 10^{0}$ and GR2

for the computational costs, Table 1 shows that already at CFL $= 1000$ the gain in CPU time obtained with the implicit scheme is large, both for 1^{st} and 2^{nd}-order of accuracy. The CPU gain obtained with the implicit scheme is significantly larger for 2^{nd}-order accuracy. Indeed, when the implicit formulation is used, there are not significant differences, in terms of CPU time, between the 1^{st} and 2^{nd}-order simulations. Instead in the explicit case an important computational cost increase is observed to reach 2^{nd}-order accuracy: the 2^{nd}-order approach is $\simeq 2.4$ times more expensive than the 1^{st}-order one. As a consequence, already at CFL $= 1000$ using 1 DeC iteration the 2^{nd}-order implicit approach is more than 60 times faster than the explicit one on GR1 and about 30 times faster on GR2. The CPU gain of the 2^{nd}-order implicit approach can be further increased considering 3 DeC iterations and CFL $= 10^4$. For the fast speed of interaction case, to avoid loss of accuracy the CFL number of the implicit scheme must be lowered down to 1. On the other hand, by increasing the number of DeC iterations, it is possible to increase the maximum

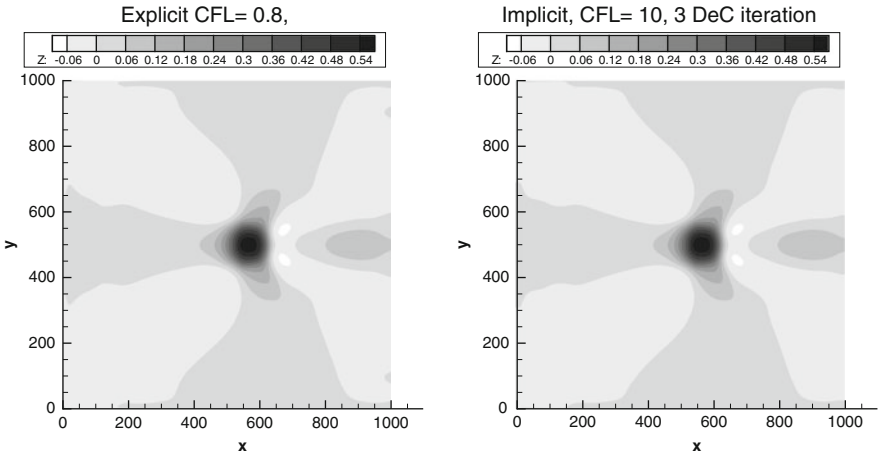

Fig. 2 Comparison of the results for the bed profile of the 2^{nd}-order scheme, $A = 1$, Grid GR2

CFL value by a factor 10. As an example Fig. 2 shows a comparison between the explicit and implicit approach at different CFL values for the grid GR2. Due to the reduced CFL number achievable without loss of accuracy for the implicit scheme 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 Table 1. Summarizing, in order 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. Also, the increase of the number of DeC iterations allows the maximum CFL number achievable without loosing in accuracy to be increased, and therefore the simulation CPU time is reduced. The implicit code has been found to be computationally more efficient than the explicit one for slow rates of the interaction between the bed and the flow.

Table 1 CPU time required for the considered simulations, comparison between explicit and implicit approach, both at first and second-order of accuracy

	$A = 0.001$			$A = 1$		
Method	GR1	GR2	CFL	GR1	GR2	CFL
Explicit 1^{st} order	12824s	103238s	0.8	21.0s	169.7s	0.8
Explicit 2^{nd} order	30996s	247215s	0.8	52.4s	409.9s	0.8
Implicit 1^{st} order	323.6s	4336s	10^3	191.5s	1541s	10^0
Implicit 2^{nd} order 1 DeC	481.5s	8537s	10^3	198.7s	1582s	10^0
Implicit 2^{nd} order 3 DeC	265.9s	4866s	10^4	74.5s	606.8s	10^1

Acknowledgements This work has been realized in the framework of the EuroMéditerranée 3 + 3 network MhyCoF.

References

1. F. Benkhaldoun, S. Sahmim, M. Seaïd. *A two-dimensional finite volume morphodynamic model on unstructured triangular grids.* Int. J. Numer. Meth. Fluids, 63:1296–1327, 2010.
2. A. Bermudez, M.E. Vazquez. *Upwind methods for hyperbolic conservation laws with source terms.* Computers & Fluids, 23(8):1049–1071, 1994.
3. S. Camarri, M.V. Salvetti, B. Koobus, A. Dervieux. *A low-diffusion MUSCL scheme for LES on unstructured grids.* Computers & Fluids, 33:1101–1129, 2004.
4. M.J. Castro Díaz, E.D. Fernández-Nieto, 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.
5. M.J. Castro Díaz, E.D. Fernández-Nieto, A.M. Ferreiro, C. Parés. *Two-dimensional sediment transport models in shallow water equations. A second order finite volume approach on unstructured meshes.* Computer Meth. Appl. Mech. Eng., 198:2520–2538, 2009.
6. A.J. Grass. *Sediments transport by waves and currents.* Tech. Rep., SERC London Cent. Mar. Technol., Report No. FL29, 1981.
7. L. Hascoët, V. Pascual. *TAPENADE 2.1 User's Guide.* Tech. Rep. n 300. INRIA, 2004.
8. J. Hudson, P. K. Sweby. *Formulations for Numerically Approximating Hyperbolic Systems Governing Sediment Transport.* J. Sci. Comput., 19:225-252, 2003.
9. B. van Leer. *Towards the ultimate conservative difference scheme* V*: a second-order sequel to* G*odunov's method.* J. Comput. Phys., 32(1):101–136, 1979.
10. R. Martin, H. Guillard. *A second order defect correction scheme for unsteady problems.* Computers & Fluids, 25(1):9–27, 1996.
11. S. Sahmim, F. Benkhaldoun, F. Alcrudo. *A sign matrix based scheme for non-homogeneous PDE's with an analysis of the convergence stagnation phenomenon.* J. Comput. Phys., 226(2):1753–1783, 2007.

The paper is in final form and no similar paper has been or is being submitted elsewhere.