# Hands-on session 3: 1D bedload transport model for erosive flow simulation

## Workshop 2 – RESCUER MSCA Doctoral Network

Sergio Martínez-Aranda
*Fluid Dynamics Tecnologies - I3A, University of Zaragoza*
sermar@unizar.es

Universidad Zaragoza
1542

Escuela de Ingeniería y Arquitectura
Universidad Zaragoza

Instituto Universitario de Investigación en Ingeniería de Aragón
Universidad Zaragoza

# Outline

## Proposed activities

1) To complete a C/C++ implementation on a 1D bedload transport model for rectangular cross-section channels.
2) To perform a sensibility analysis in order to identify the most relevant parameters for the bedload model.
3) To apply the 1D bedload transport model for the estimation of equilibrium states in fluvial regimes.

## Available materials

1) Hands-on session summary (this document).
2) sed1D_simulator code, based on the previous swe1D_simulator
3) Practice 1 guide + pre-configured input files.
4) Practice 2 guide + pre-configured input files.

# 1D bedload transport model

## Governing equation for rectangular cross-section channel

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A} + \frac{1}{2} g \frac{A^2}{B}\right) = -gA\frac{\partial z_b}{\partial x} - gA\frac{n^2\,|u|\,u}{h^{4/3}}$$

$$\frac{\partial A_b}{\partial t} + \frac{1}{1-p}\frac{\partial Q_b}{\partial x} = 0$$

Hydrodynamic primitive variables

$$A = Bh \quad Q = Bhu$$

Bedload primitive variables

$$A_b = B(z_b - z_r) \quad Q_b = B\,q_b$$

## Bedload transport rate

1) Fixed-bed $\mathrm{q_b} = 0$

2) Grass formula $\mathrm{q_b} = G\,u^2\,u$

3) MPM formula $\mathrm{q_b} = \beta_B\;8(\theta - \theta_c)^{3/2}\sqrt{(\rho_s/\rho_w - 1)gd_s^3}$

$$\theta = \frac{\tau_b}{(\rho_s - \rho_w)gd_s}$$ Shields stress

$$\tau_b = \rho_w g \frac{n^2\,u^2}{h^{1/3}}$$ Bed shear stress

# Split method for 1D bedload model

Updating conservative bed area at cells

$$(A_b)_i^{n+1} = (A_b)_i^n - \frac{\Delta t}{\Delta x}\frac{1}{1-p}\left[(Q_b)_{i+1/2}^{\downarrow -} - (Q_b)_{i-1/2}^{\downarrow -}\right]$$

Computing bedload rate at intercell edges

$$(Q_b)_{i+1/2}^{\downarrow -} = \begin{cases} (Q_b)_i^n & \text{if} \quad (\widetilde{\lambda}_b)_{i+1/2} > 0 \\ (Q_b)_{i+1}^n & \text{if} \quad (\widetilde{\lambda}_b)_{i+1/2} < 0 \end{cases}$$

Computing virtual bedload celerity at intercell edges

$$(\widetilde{\lambda}_b)_{i+1/2} = \frac{1}{1-p}\frac{(q_b)_{i+1}^n - (q_b)_i^n}{(z_b)_{i+1}^n - (z_b)_i^n}$$

Reducing time step due to bedload transport

$$\Delta t_b = \min_k \left(\frac{\Delta x}{|\widetilde{\lambda}_b|_{i+1/2}}\right)$$

# C/C++ code implementation

```
h_compute_initial_flow_variables( nCells,
h_compute_initial_bedload_variables( nCells,

while(t <= simTime) {

        h_compute_flow_time_step( nWalls,
        h_compute_bedload_time_step( nWalls,
            //**** complete code here ********

        h_compute_wall_fluxes( nWalls,
        h_compute_bedload_wall_fluxes(nWalls,
            //**** complete code here ********

        h_update_cells( nCells,
        h_update_bedload_cells( nCells,
            //**** complete code here ********

        h_set_inlet_conditions( nCells,
        h_set_bedload_inlet_conditions( nCells,

}
```

$$(\widetilde{\lambda}_b)_{i+1/2} = \frac{1}{1-p}\frac{(q_b)^n_{i+1} - (q_b)^n_i}{(z_b)^n_{i+1} - (z_b)^n_i}$$

$$\Delta t_b = \min_k \left( \frac{\Delta x}{|\widetilde{\lambda}_b|_{i+1/2}} \right)$$

$$(Q_b)^{\downarrow -}_{i+1/2} = \begin{cases} (Q_b)^n_i & \text{if} \quad (\widetilde{\lambda}_b)_{i+1/2} > 0 \\ (Q_b)^n_{i+1} & \text{if} \quad (\widetilde{\lambda}_b)_{i+1/2} < 0 \end{cases}$$

$$(A_b)^{n+1}_i = (A_b)^n_i - \frac{\Delta t}{\Delta x}\frac{1}{1-p}\left[ (Q_b)^{\downarrow -}_{i+1/2} - (Q_b)^{\downarrow -}_{i-1/2} \right]$$

$$(Q_b)^{n+1}_i = f(h,u)$$