A multiscale method applied to shallow water flow

Guillaume Chiavassa[1], Rosa Donat[2], Anna Martínez-Gavara[2]

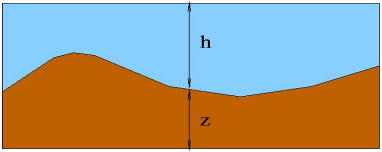
[1]Ecole Centrale de Marseille.
[2]Universitat de València.

Benasque
September, 2007

Outline

Shallow water equations

- hyperbolic system of conservation laws
- source terms are due to topography
- not considering wind effects nor Coriolis force.



$$U_t + F(U)_x + E(U)_y = S$$

$$\begin{pmatrix} h \\ q_1 \\ q_2 \end{pmatrix}_t + \begin{pmatrix} q_1 \\ \frac{q_1^2}{h} + \frac{1}{2}gh^2 \\ \frac{q_1 q_2}{h} \end{pmatrix}_x + \begin{pmatrix} q_2 \\ \frac{q_1 q_2}{h} \\ \frac{q_2^2}{h} + \frac{1}{2}gh^2 \end{pmatrix}_y = \begin{pmatrix} 0 \\ -ghz_x \\ -ghz_y \end{pmatrix}$$

Numerical Treatment

- Fractional step methods do not respect steady/quasy-steady states.

  [Leveque]
- Source term upwinding.

  [Roe,Bermúdez-Vázquez]
- Follow approach of

  [Caselles-Donat-Haro, Donat-Marquina, Gascón-Corberán]

Numerical Treatment

- Fractional step methods do not respect steady/quasy-steady states.

  [Leveque]
- Source term upwinding.

  [Roe,Bermúdez-Vázquez]
- Follow approach of

  [Caselles-Donat-Haro, Donat-Marquina, Gascón-Corberán]

Numerical Treatment

- Fractional step methods do not respect steady/quasy-steady states.

  [Leveque]
- Source term upwinding.

  [Roe,Bermúdez-Vázquez]
- Follow approach of

  [Caselles-Donat-Haro, Donat-Marquina, Gascón-Corberán]

The numerical scheme

$U_t + (F+B)_x + (E+C)_y = 0$

FORMAL FLUXES = physical fluxes $+ \begin{cases} B(x,y,t) = \left(0, \int_{\bar{x}}^{x} ghz_x ds, 0\right)^T \\ C(x,y,t) = \left(0, 0, \int_{\bar{y}}^{y} ghz_y ds\right)^T \end{cases}$

NUMERICAL METHOD (Shu-Osher, Finite Difference framework)

- TVD-Runge-Kutta method (time integration).
- dimension by dimension discretization.
- ENO reconstruction of formal fluxes:
  - characteristic speeds of physical fluxes.
  - trapezoidal rule (integral approximation).

The numerical scheme

$U_t + (F+B)_x + (E+C)_y = 0$

$$\text{FORMAL FLUXES} = \text{physical fluxes} + \begin{cases} B(x,y,t) = \left(0, \int_{\bar{x}}^{x} ghz_x ds, 0\right)^T \\ C(x,y,t) = \left(0, 0, \int_{\bar{y}}^{y} ghz_y ds\right)^T \end{cases}$$

NUMERICAL METHOD (Shu-Osher, Finite Difference framework)

- TVD-Runge-Kutta method (time integration).
- dimension by dimension discretization.
- ENO reconstruction of formal fluxes:
  - characteristic speeds of physical fluxes.
  - trapezoidal rule (integral approximation).

The numerical scheme

$U_t + (F+B)_x + (E+C)_y = 0$

$$\text{FORMAL FLUXES} = \text{physical fluxes} + \begin{cases} B(x,y,t) = \left(0, \int_{\bar{x}}^{x} ghz_x ds, 0\right)^T \\ C(x,y,t) = \left(0, 0, \int_{\bar{y}}^{y} ghz_y ds\right)^T \end{cases}$$

NUMERICAL METHOD (Shu-Osher, Finite Difference framework)

- TVD-Runge-Kutta method (time integration).
- dimension by dimension discretization.
- ENO reconstruction of formal fluxes:
  - characteristic speeds of physical fluxes.
  - trapezoidal rule (integral approximation).

The numerical scheme

$U_t + (F + B)_x + (E + C)_y = 0$

$$\text{FORMAL FLUXES} = \text{physical fluxes} + \begin{cases} B(x,y,t) = \left(0, \int_{\bar{x}}^{x} ghz_x ds, 0\right)^T \\ C(x,y,t) = \left(0, 0, \int_{\bar{y}}^{y} ghz_y ds\right)^T \end{cases}$$

NUMERICAL METHOD (Shu-Osher, Finite Difference framework)

- TVD-Runge-Kutta method (time integration).
- dimension by dimension discretization.
- ENO reconstruction of formal fluxes:
  - characteristic speeds of physical fluxes.
  - trapezoidal rule (integral approximation).

The semi-discrete formulation 1D

[CaDoHa] V. Caselles, R. Donat, G. Haro, *Flux Gradient and Source Term Balancing for certain high resolution shock capturing schemes*. Submitted.

$$\frac{dU}{dt} + Div(U) = 0$$

$$\frac{G^{+}_{i+\frac{1}{2}} - G^{-}_{i-\frac{1}{2}}}{\triangle x}$$

$G^{\pm}_{i+\frac{1}{2}}$ involve quantities: $B_{i,i+1} = \int_{x_i}^{x_{i+1}} ghz_x dx$.

The semi-discrete formulation 1D

[CaDoHa] V. Caselles, R. Donat, G. Haro, *Flux Gradient and Source Term Balancing for certain high resolution shock capturing schemes*. Submitted.

$$\frac{dU}{dt} + Div(U) = 0$$

$$\frac{G^{+}_{i+\frac{1}{2}} - G^{-}_{i-\frac{1}{2}}}{\triangle x}$$

$G^{\pm}_{i+\frac{1}{2}}$ involve quantities: $B_{i,i+1} = \int_{x_i}^{x_{i+1}} ghz_x dx$.

The semi-discrete formulation 1D

[CaDoHa] V. Caselles, R. Donat, G. Haro, *Flux Gradient and Source Term Balancing for certain high resolution shock capturing schemes*. Submitted.

$$\frac{dU}{dt} + Div(U) = 0$$

$$\frac{G^{+}_{i+\frac{1}{2}} - G^{-}_{i-\frac{1}{2}}}{\triangle x}$$

$G^{\pm}_{i+\frac{1}{2}}$ involve quantities: $B_{i,i+1} = \int_{x_i}^{x_{i+1}} g h z_x dx$.

The semi-discrete formulation 1D

[CaDoHa] V. Caselles, R. Donat, G. Haro, *Flux Gradient and Source Term Balancing for certain high resolution shock capturing schemes*. Submitted.

$$\frac{dU}{dt} + Div(U) = 0$$

$$\frac{G^{+}_{i+\frac{1}{2}} - G^{-}_{i-\frac{1}{2}}}{\triangle x}$$

$G^{\pm}_{i+\frac{1}{2}}$ involve quantities: $B_{i,i+1} = \int_{x_i}^{x_{i+1}} g h z_x dx$.

C-PROPERTY. quiescent flow: $h = constant - z \quad q_1 = q_2 = 0$

- exact C-property $\Longrightarrow$ numerical scheme exact.
- approximate C-property $\Longrightarrow$ numerical scheme $\mathcal{O}(\triangle x^2)$.

The semi-discrete formulation 1D

[CaDoHa] V. Caselles, R. Donat, G. Haro, *Flux Gradient and Source Term Balancing for certain high resolution shock capturing schemes*. Submitted.

$$\frac{dU}{dt} + Div(U) = 0$$

$$\frac{G^{+}_{i+\frac{1}{2}} - G^{-}_{i-\frac{1}{2}}}{\triangle x}$$

$G^{\pm}_{i+\frac{1}{2}}$ involve quantities: $B_{i,i+1} = \int_{x_i}^{x_{i+1}} g h z_x dx$.

C-PROPERTY. quiescent flow: $h = constant - z \quad q_1 = q_2 = 0$

- exact C-property $\Longrightarrow$ numerical scheme exact.
- approximate C-property $\Longrightarrow$ numerical scheme $\mathcal{O}(\triangle x^2)$.

The semi-discrete formulation

$$G^{\pm}_{i+\frac{1}{2}} = \sum_{p=1}^{2} (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^L R^p(U^L) + (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^R R^p(U^R)$$

The semi-discrete formulation

$$G^{\pm}_{i+\frac{1}{2}} = \sum_{p=1}^{2} (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^L R^p(U^L) + (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^R R^p(U^R)$$

- If $\lambda^p(U^L_{i+\frac{1}{2}}) > 0$ and $\lambda^p(U^R_{i+\frac{1}{2}}) > 0$: **upwind from the left** $(\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^R = 0$

  $(\tilde{G}^{p,+}_{i+\frac{1}{2}})^L = L^p(U^L) \cdot F_i + HOT^L_{i+\frac{1}{2}}$

  $(\tilde{G}^{p,-}_{i+\frac{1}{2}})^L = L^p(U^L) \cdot (F_i - B_{i,i+1}) + HOT^L_{i+\frac{1}{2}}$

The semi-discrete formulation

$$G^{\pm}_{i+\frac{1}{2}} = \sum_{p=1}^{2} (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^L R^p(U^L) + (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^R R^p(U^R)$$

- If $\lambda^p(U^L_{i+\frac{1}{2}}) < 0$ and $\lambda^p(U^R_{i+\frac{1}{2}}) < 0$: **upwind from the right** $(\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^L = 0$

  $(\tilde{G}^{p,+}_{i+\frac{1}{2}})^R = L^p(U^R) \cdot (F_{i+1} + B_{i,i+1}) + HOT^R_{i+\frac{1}{2}}$

  $(\tilde{G}^{p,-}_{i+\frac{1}{2}})^R = L^p(U^R) \cdot F_{i+1} + HOT^R_{i+\frac{1}{2}}$

The semi-discrete formulation

$$G^{\pm}_{i+\frac{1}{2}} = \sum_{p=1}^{2} (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^L R^p(U^L) + (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^R R^p(U^R)$$

- If $\lambda^p(U^L_{i+\frac{1}{2}}) * \lambda^p(U^R_{i+\frac{1}{2}}) < 0$: **sonic point nearby** $\alpha = \text{máx}(|\lambda^p(U^L_{i+\frac{1}{2}})|, |\lambda^p(U^R_{i+\frac{1}{2}})|)$

$$(\tilde{G}^{p,+}_{i+\frac{1}{2}})^L = \frac{1}{2} L^p(U^L) \cdot (F_i + \alpha U_i) + HOT^L_{i+\frac{1}{2}}$$

$$(\tilde{G}^{p,-}_{i+\frac{1}{2}})^L = \frac{1}{2} L^p(U^L) \cdot (F_i + \alpha U_i - B_{i,i+1}) + HOT^L_{i+\frac{1}{2}}$$

$$(\tilde{G}^{p,+}_{i+\frac{1}{2}})^R = \frac{1}{2} L^p(U^R) \cdot (F_{i+1} - \alpha U_{i+1} + B_{i,i+1}) + HOT^R_{i+\frac{1}{2}}$$

$$(\tilde{G}^{p,-}_{i+\frac{1}{2}})^R = \frac{1}{2} L^p(U^R) \cdot (F_{i+1} - \alpha U_{i+1}) + HOT^R_{i+\frac{1}{2}}$$

The semi-discrete formulation

$$G^{\pm}_{i+\frac{1}{2}} = \sum_{p=1}^{2} (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^L R^p(U^L) + (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^R R^p(U^R)$$

- 1J scheme $\Longrightarrow U^* = (U^L + U^R)/2 \Longrightarrow$ exact C-property.
- 2J scheme $\Longrightarrow U^L \neq U^R \Longrightarrow$ approximate C-property ($r \geq 2$).
- 1J-2J scheme $\Longrightarrow$ get the benefits of both (our choice).

The semi-discrete formulation

$$G^{\pm}_{i+\frac{1}{2}} = \sum_{p=1}^{2} (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^{L} R^{p}(U^{L}) + (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^{R} R^{p}(U^{R})$$

- 1J scheme $\Longrightarrow U^{*} = (U^{L} + U^{R})/2 \Longrightarrow$ exact C-property.
- 2J scheme $\Longrightarrow U^{L} \neq U^{R} \Longrightarrow$ approximate C-property ($r \geq 2$).
- 1J-2J scheme $\Longrightarrow$ get the benefits of both (our choice).

The semi-discrete formulation

$$G^{\pm}_{i+\frac{1}{2}} = \sum_{p=1}^{2} (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^{L} R^{p}(U^{L}) + (\tilde{G}^{p,\pm}_{i+\frac{1}{2}})^{R} R^{p}(U^{R})$$

- 1J scheme $\Longrightarrow U^{*} = (U^{L} + U^{R})/2 \Longrightarrow$ exact C-property.
- 2J scheme $\Longrightarrow U^{L} \neq U^{R} \Longrightarrow$ approximate C-property ($r \geq 2$).
- 1J-2J scheme $\Longrightarrow$ get the benefits of both (our choice).

General framework $2D$

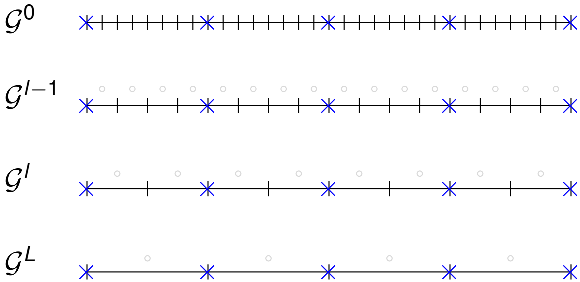
Goal

Reduce the CPU time

Means

Analyze smoothness using Harten's interpolatory. Multiresolution transform. [Chiavassa-Donat, SISC01]

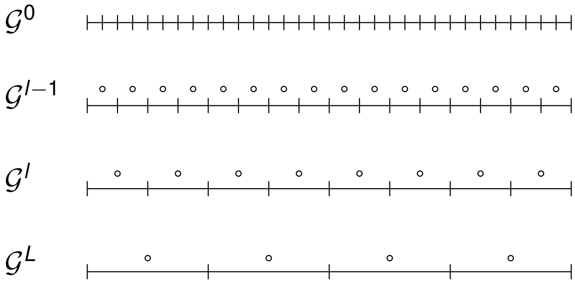
Algorithm



Interpolatory multiresolution

- $\{\mathcal{G}^l, l = 0, \ldots, L\}$: $(x_i, y_j) \in \mathcal{G}^l \iff (x_{2^l i}, y_{2^l j}) \in \mathcal{G}^0$.
- $(d^l_{ij})_{l,i,j}$(wavelet coefficients) USED TO DETERMINE $(b^l_{ij})_{l,i,j}$(marker).
  - $(u^0_{i,j})_{i,j}$ on $\mathcal{G}^0 \Longrightarrow u^l_{i,j} = u^0_{2^l i, 2^l j}$ on $\mathcal{G}^l$
  - $d^l_{i,j} = u^{l-1}_{i,j} - \mathcal{I}[(x_i, y_j); u^l] \quad (x_i, y_j) \in \mathcal{G}^{l-1} \backslash \mathcal{G}^l$

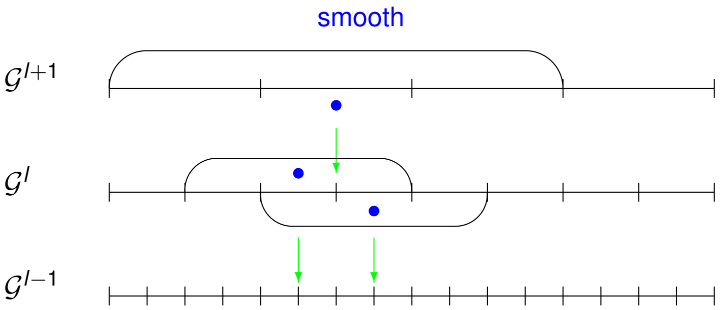
Algorithm



Interpolatory multiresolution

- $\{\mathcal{G}^l, l = 0, \ldots, L\}$: $(x_i, y_j) \in \mathcal{G}^l \iff (x_{2^l i}, y_{2^l j}) \in \mathcal{G}^0$.
- $(d^l_{ij})_{l,i,j}$(wavelet coefficients) USED TO DETERMINE $(b^l_{ij})_{l,i,j}$(marker).
  - $(u^0_{i,j})_{i,j}$ on $\mathcal{G}^0 \Longrightarrow u^l_{i,j} = u^0_{2^l i, 2^l j}$ on $\mathcal{G}^l$
  - $d^l_{i,j} = u^{l-1}_{i,j} - \mathcal{I}[(x_i, y_j); u^l] \quad (x_i, y_j) \in \mathcal{G}^{l-1} \backslash \mathcal{G}^l$

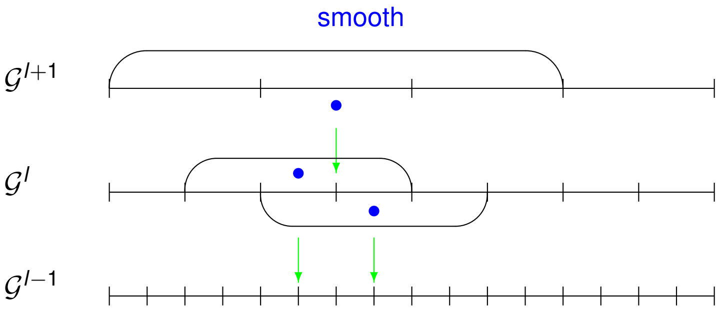
Algorithm



Interpolatory multiresolution

- $\{\mathcal{G}^l, l = 0, \ldots, L\}$: $(x_i, y_j) \in \mathcal{G}^l \iff (x_{2^l i}, y_{2^l j}) \in \mathcal{G}^0$.
- $(d^l_{ij})_{l,i,j}$(wavelet coefficients) USED TO DETERMINE $(b^l_{ij})_{l,i,j}$(marker).
  - $(u^0_{i,j})_{i,j}$ on $\mathcal{G}^0 \Longrightarrow u^l_{i,j} = u^0_{2^l i, 2^l j}$ on $\mathcal{G}^l$
  - $d^l_{i,j} = u^{l-1}_{i,j} - \mathcal{I}[(x_i, y_j); u^l] \quad (x_i, y_j) \in \mathcal{G}^{l-1} \backslash \mathcal{G}^l$

Algorithm



Interpolatory multiresolution

- $\{\mathcal{G}^l, l = 0, \dots, L\}$: $(x_i, y_j) \in \mathcal{G}^l \iff (x_{2^l i}, y_{2^l j}) \in \mathcal{G}^0$.
- $(d^l_{ij})_{l,i,j}$(wavelet coefficients) USED TO DETERMINE $(b^l_{ij})_{l,i,j}$(marker).
  - $(u^0_{i,j})_{i,j}$ on $\mathcal{G}^0 \Longrightarrow u^l_{i,j} = u^0_{2^l i, 2^l j}$ on $\mathcal{G}^l$
  - $d^l_{i,j} = u^{l-1}_{i,j} - \mathcal{I}[(x_i, y_j); u^l] \quad (x_i, y_j) \in \mathcal{G}^{l-1} \backslash \mathcal{G}^l$

Algorithm

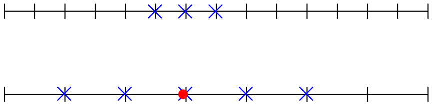
A thresholding algorithm

$l = L, \ldots, 1$

$$|d_{ij}^l| \geq \varepsilon \qquad \Longrightarrow b_{i-k,j-m}^l = 1 \quad k, m = -2, \cdots, 2$$
$$|d_{ij}^l| \geq 2^r \varepsilon \quad \text{and} \quad l > 1 \quad \Longrightarrow b_{2i-k,2j-m}^{l-1} = 1 \quad k, m = -1, 0, 1$$

b_{2i}^{l-1}

b_i^l



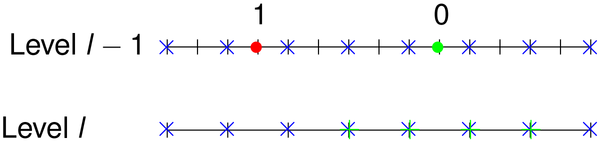
Algorithm

Numerical divergence evaluate on the coarsest grid $\mathcal{G}^L$

Multilevel evaluation

$l = L, \ldots, 1$

$$b^l_{ij} = 1, \quad \text{compute } Div^{l-1}(U)_{ij} \text{ directly with the scheme}$$
$$b^l_{ij} = 0, \quad \text{compute } Div^{l-1}(U)_{ij} = \mathcal{I}[(x_i, y_j); Div^l(U)]$$



Evaluation of the algorithm: Quality and Efficiency

Quality

$$\frac{\| h^n_{mr} - h^n_{ref} \|_{\ell_1}}{\| h^n_{ref} \|_{\ell_1}}$$

Efficiency

- $\%f$ percentage of numerical divergence computed.
- θ cpu gain

$$\theta = \frac{\text{CPU time for reference computation}}{\text{CPU time for multilevel computation}}$$

Evaluation of the algorithm: Quality and Efficiency

Quality

$$\frac{\| h^n_{mr} - h^n_{ref} \|_{\ell_1}}{\| h^n_{ref} \|_{\ell_1}}$$

Efficiency

- $\% f$ percentage of numerical divergence computed.
- θ cpu gain

$$\theta = \frac{\text{CPU time for reference computation}}{\text{CPU time for multilevel computation}}$$

Evaluation of the algorithm: Quality and Efficiency

Quality

$$\frac{\| h^n_{mr} - h^n_{ref} \|_{\ell_1}}{\| h^n_{ref} \|_{\ell_1}}$$

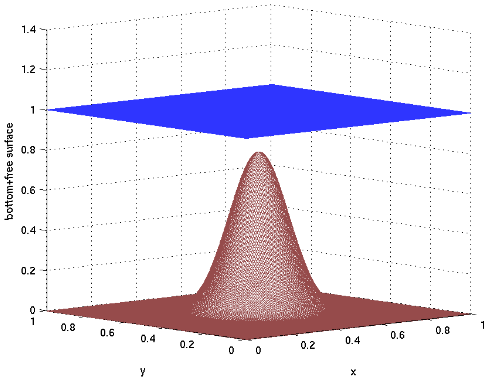
Efficiency

- $\%f$ percentage of numerical divergence computed.
- θ cpu gain

$$\theta = \frac{\text{CPU time for reference computation}}{\text{CPU time for multilevel computation}}$$

The C-property

Grid size $\mathcal{G}^0$	$\%f_{min}$ - $\%f_{max}$	l_1-error
257×257	6,5784 - 6,5784	$5{,}4674 \cdot 10^{-15}$
513×513	1,6510 - 1,6510	$1{,}1376 \cdot 10^{-14}$

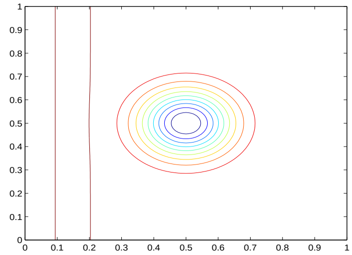


2-D Test

Initial Data

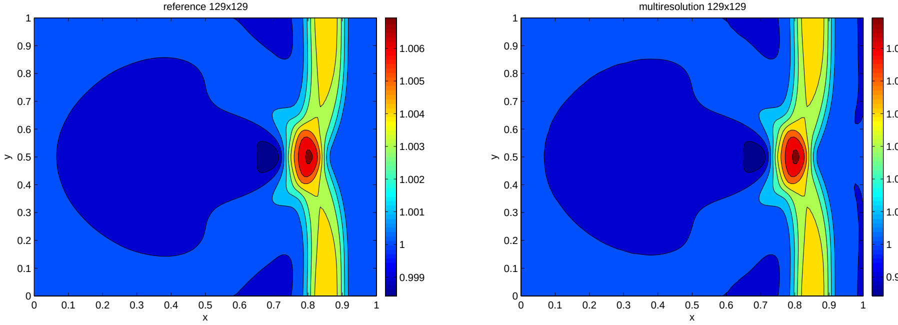
$$z(x,y) = 0{,}5e^{-50((x-0{,}5)^2+(y-0{,}5)^2)} \qquad q_1(x,y) = 0$$

$$h(x,y) = \begin{cases} 1{,}01 - z(x,y), & 0{,}1 < x < 0{,}2; \\ 1 - z(x,y), & \textit{otherwise.} \end{cases} \qquad q_2(x,y) = 0$$



2-D Test

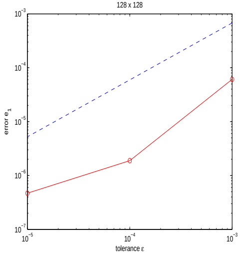
REFERENCE SIMULATION($t = 0{,}7$) MULTILEVEL SIMULATION ($\varepsilon = 10^{-4}$)



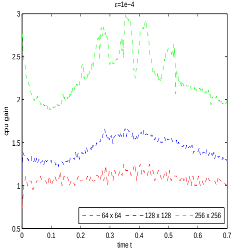
Evaluation of the algorithm: Efficiency

Grid size $\mathcal{G}^0$	$\%f_{min}$	-	$\%f_{max}$	cpu gain θ
64×64	81.92	-	93.80	1.0778
128×128	54.55	-	74.64	1.4220
256×256	27.49	-	46.52	2.2592

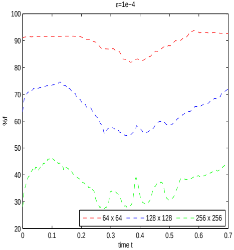
Tolerance versus error



Time evolution of θ



Time evolution of %f



Future work

- Simulation of water avalanches or dam-breaks over dry beds with variable topography.
- Simulations on real topographies.
- Consider source terms due to wind effects and/or Coriolis force.