GPU implementation of coupled models of bed load transport and multi-layered shallow water model

Mohamed Boubekeur, Fayssal Benkhaldoun

LAGA, Université Paris 13, France

03-07-2018

Summary

GPU implementation of coupled models of bed load transport and multi-layered shallow water model

Inroduction :

- Modelling water flows is based on the formulation of the appropriate equations of continuity and motion of water.
- Free-surface flows represent a three-dimensional turbulent Newtonian flow :
  - Complicated geometrical domains.
  - Moving boundaries.
- High computational cost.

Introduction

- The shallow water models.
  - environmental and hydraulic engineering : tidal flows in an estuary or coastal regions, rivers, reservoir and open channel.
- Multi-layered shallow water models.
  - avoid the expensive three-dimensional Navier-Stokes equations.
  - obtain stratified horizontal flow velocities as the pressure distribution is nearly hydrostatic.
- Higher is the number of layer, better is.
- Increase the computational cost.
- Need of HPC Resources.
- A cost effective way is using Graphics Processor Units.
- CUDA implementation for NVIDA card's.

The shallow water model

- The equations of shallow water flows are :

$$
\begin{aligned}
\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) &= 0 \\
\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(huv) &= -gh\frac{\partial Z}{\partial x} + f_c hv \\
\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) &= -gh\frac{\partial Z}{\partial y} - f_c hu
\end{aligned}
$$

where

- (u, v) the velocity field.
- h is the water height.
- Z is the topography of the bed.
- g is the gravity.
- f_c is the Coriolis forces.

Conservative form

- The conservative form of the equations is :

$$\frac{\partial W}{\partial t} + \frac{\partial F(W)}{\partial x} + \frac{\partial G(W)}{\partial y} = Q(W) + R(W).$$

where W is the vector of conserved variable, F and G the vectors of flux functions, Q and R are the vector of source terms.

$$W = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad F(W) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \quad G(W) = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix},$$

$$Q(W) = \begin{pmatrix} 0 \\ -gh\frac{\partial Z}{\partial x} \\ -gh\frac{\partial Z}{\partial y} \end{pmatrix}, \quad R(W) = \begin{pmatrix} 0 \\ f_c hv \\ f_c hu \end{pmatrix},$$

Numerical scheme

- A hybrid finite volume-characteristics scheme (FVC) introduced by F. Benkhaldoun and M. Seaid in 2010.
  - Predictor-corrector.
  - Avoid the solution of Riemann problems during the time integration.
  - Use the method of characteristics to construct the numerical fluxes.
  - well balanced, conservative, non-oscillatory.
  - Easily parallelizable.

Numerical scheme

- The spatial domain is descritized into control volumes $C_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ with uniform sizes.
- The semi-discrete equation is :

$$\frac{\mathrm{d}W_{i,j}}{\mathrm{d}t} + \frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} + \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y} = Q_{i,j} + R_{i,j}$$

- $F_{i\pm\frac{1}{2},j} = F(W_{i\pm\frac{1}{2},j})$, $G_{i,j\pm\frac{1}{2}} = G(W_{i,j\pm\frac{1}{2}})$ are the numerical fluxes.
- They are obtained by the FVC method.
- $Q_{i,j} = Q(W_{i,j})$ and $R_{i,j} = R(W_{i,j})$ are the source terms.
- We adopt the following notations :

$$W_{i\pm\frac{1}{2},j}(t) = W(t, x_{i\pm\frac{1}{2}}, y_j)$$

$$W_{i,j\pm\frac{1}{2}}(t) = W(t, x_i, y_{j\pm\frac{1}{2}})$$

$$W_{i,j} = \frac{1}{\Delta x}\frac{1}{\Delta y}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} W(t,x,y)\mathrm{d}y\mathrm{d}x$$

Numerical scheme

- The time discretization is an explicit Euler scheme

$$W_{i,j}^{n+1} = W_{i,j}^{n} - \Delta t \frac{F_{i+\frac{1}{2},j}^{n} - F_{i-\frac{1}{2},j}^{n}}{\Delta x} - \Delta t \frac{G_{i,j+\frac{1}{2}}^{n} - G_{i,j-\frac{1}{2}}^{n}}{\Delta y} + \Delta t Q_{i,j}^{n} + \Delta t R_{i,j}^{n}$$

- The CFL condition C is specified as :

$$\Delta t = C \frac{\min(\Delta x, \Delta y)}{\max(\lambda, \mu)}$$

where C is the Courant number to be chosen less than unity, λ and μ are the maximum of eigenvalues associated to the model defined as :

$$\lambda = \max(|u + \sqrt{gh}|, |u|, |u - \sqrt{gh}|)$$
$$\mu = \max(|v + \sqrt{gh}|, |v|, |v - \sqrt{gh}|)$$

CUDA implementation

- Considering each thread as a volume control.
- The code architecture :
  - Compute the fluxes :
    - The predictor step.
    - The corrector step.
  - Compute W^{n+1}.
- Draw the water height H with OpenGL.
- minimize communication between CPU and GPU

Numerical result

- $\Omega = [-10, 10] \times [-10, 10]$.
- Neumann boundary conditions.
- Initial condition.

$$h(0, x, y) = \begin{cases} 3 & \text{if } x < 0. \\ 1 & \text{elsewhere.} \end{cases}$$

$$u(0, x, y) = 0$$
$$v(0, x, y) = 0$$

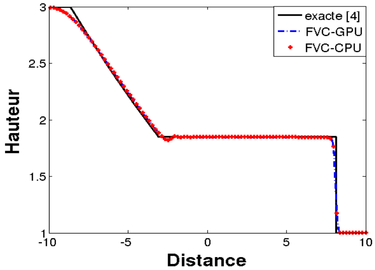
GPU vs CPU

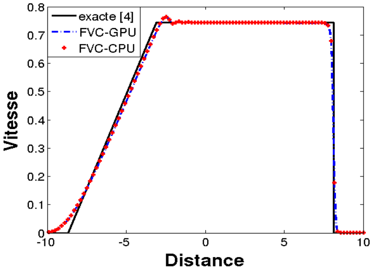
- CPU : Intel(R) Core(TM) i7-4930MX CPU @ 3.00GHz
- GPU : NVIDIA Quadro K5100M

mesh	GPU	CPU	
		8 cores	1 core
50×50	0.02	0.07	0.16
100×100	0.44	0.59	1.31
200×200	1.33	4.13	1.78
400×400	6.06	26.71	87.21
600×600	18.04	88.12	293.61
800×800	37.73	215.15	693.89
1000×1000	66.43	383.71	1361.25

Execution times in seconds obtained for $t_{end} = 5s$

Water heights and velocity fields obtained for $t = 10s$





Water heights (left) and velocity fields (right) obtained for dam-break problem $t = 10s$

Numerical result

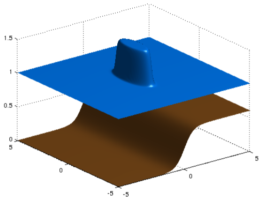
- $\Omega = [-5,5] \times [-5,5]$.
- Neumann boundary conditions.
- Initial condition.

$$h(0,x,y) = 1 + \frac{1}{4}\left(1 - \tanh\left(\frac{\sqrt{ax^2+by^2}-1}{c}\right)\right)$$
$$u(0,x,y) = 0$$
$$v(0,x,y) = 0$$

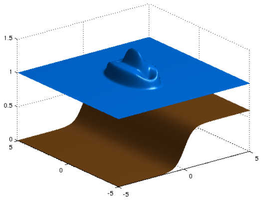
where $a = 52$, $b = 25$ and $c = 0.1$.

- The gravity $g = 1$ and the Coriolis $f_c = 1$.

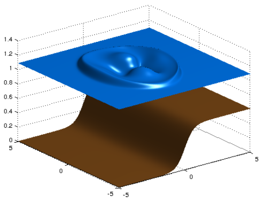
Free surface evolution



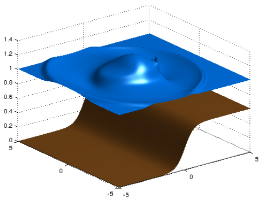
(a) $t = 0$



(b) $t = 0.5$



(c) $t = 2$



(d) $t = 4$

The multi-layer shallow water model

- The equations for multi-layered shallow water flows with mass exchange are :

$$\frac{\partial h}{\partial t} + \sum_{\alpha=1}^{n} \frac{\partial}{\partial x}\left(l_\alpha h u_\alpha\right) + \sum_{\alpha=1}^{n} \frac{\partial}{\partial y}\left(l_\alpha h v_\alpha\right) = 0$$

$$\frac{\partial}{\partial t}\left(l_\alpha h u_\alpha\right) + \sum_{\alpha=1}^{n} \frac{\partial}{\partial x}\left(l_\alpha h u_\alpha^2 + \frac{1}{2} g l_\alpha h^2\right) + \sum_{\alpha=1}^{n} \frac{\partial}{\partial y}\left(l_\alpha h u_\alpha v_\alpha\right) = -g l_\alpha h \frac{\partial Z}{\partial x} + F_\alpha$$

$$\frac{\partial}{\partial t}\left(l_\alpha h v_\alpha\right) + \sum_{\alpha=1}^{n} \frac{\partial}{\partial x}\left(l_\alpha h u_\alpha v_\alpha\right) + \sum_{\alpha=1}^{n} \frac{\partial}{\partial y}\left(l_\alpha h v_\alpha^2 + \frac{1}{2} g l_\alpha h^2\right) = -g l_\alpha h \frac{\partial Z}{\partial y} + G_\alpha$$

with

$$\begin{aligned} F_\alpha &= F_u + F_b + F_w + F_\mu \\ G_\alpha &= G_v + G_b + G_w + G_\mu \end{aligned}$$

and

$$l_\alpha > 0, \qquad \sum_{\alpha=1}^{M} l_\alpha = 1$$

Some parameters definition

- The advection terms F_u and G_v are given by :

$$F_u = u_{\alpha+\frac{1}{2}}\xi^u_{\alpha+\frac{1}{2}} - u_{\alpha-\frac{1}{2}}\xi^u_{\alpha-\frac{1}{2}}$$
$$G_v = v_{\alpha+\frac{1}{2}}\xi^v_{\alpha+\frac{1}{2}} - v_{\alpha-\frac{1}{2}}\xi^v_{\alpha-\frac{1}{2}}$$

where the mass exchange term $\xi^u_{\alpha+\frac{1}{2}}$ can be computed as :

$$\xi^u_{\alpha+\frac{1}{2}} = \sum_{\beta=1}^{\alpha}\left(\frac{\partial(l_\beta h u_\beta)}{\partial x} - l_\beta \sum_{\gamma=1}^{n}\frac{\partial(l_\gamma h u_\gamma)}{\partial x}\right)$$

and the interface velocity is computed by a simple upwinding following the sign of the mass exchange term as :

$$u_{\alpha+\frac{1}{2}} = \begin{cases} u_\alpha, & \text{if } \xi^u_{\alpha+\frac{1}{2}} \geqslant 0 \\ u_{\alpha+1}, & \text{if } \xi^u_{\alpha+\frac{1}{2}} < 0 \end{cases}$$

Some parameters definition

- The vertical kinematic eddy viscosity terms F_μ and G_μ take into account the friction between neighbouring layers and they are defined as :

$$F_\mu = 2\nu(1-\delta_{n\alpha})\frac{u_{\alpha+1}-u_\alpha}{(l_{\alpha+1}+l_\alpha)h} - 2\nu(1-\delta_{1\alpha})\frac{u_\alpha - u_{\alpha-1}}{(l_\alpha+l_{\alpha-1})h}$$

$$G_\mu = 2\nu(1-\delta_{n\alpha})\frac{v_{\alpha+1}-v_\alpha}{(l_{\alpha+1}+l_\alpha)h} - 2\nu(1-\delta_{1\alpha})\frac{v_\alpha - v_{\alpha-1}}{(l_\alpha+l_{\alpha-1})h}$$

ν is the eddy viscosity and $\delta_{k\alpha}$ represents the Kronecker symbol.

- The external friction terms are given by :

$$F_b = -\delta_{1\alpha}\frac{\zeta_b}{\rho}$$

$$G_b = -\delta_{1\alpha}\frac{\eta_b}{\rho}$$

$$F_w = \delta_{n\alpha}\frac{\zeta_w}{\rho}$$

$$G_w = \delta_{n\alpha}\frac{\eta_w}{\rho}$$

with ρ is the water density, ζ_b and η_b are the bed shear stress, ζ_w and η_w are the shear of the blowing wind.

Numerical scheme

- The time discretization is done with an operator splitting method.
- The first step is :

$$W_{i,j}^{*} = W_{i,j}^{n} + \Delta t R_{i,j}^{n}$$

with $R_{i,j}^{n} = R_{i,j}(t_n)$.

- The second steps is :

$$W_{i,j}^{n+1} = W_{i,j}^{*} - \Delta t \frac{F_{i+\frac{1}{2},j}^{*} - F_{i-\frac{1}{2},j}^{*}}{\Delta x} - \Delta t \frac{G_{i,j+\frac{1}{2}}^{*} - G_{i,j-\frac{1}{2}}^{*}}{\Delta \partial y} + \Delta t Q_{i,j}^{*}$$

- the fluxes are computed using FVC method.
- The time integration scheme is explicit, The CFL condition C is specified as :

$$\Delta t = C \frac{\min(\Delta x, \Delta y)}{\max(\lambda, \mu)}$$

where C is the Courant number to be chosen less than unity, λ and μ are the maximum of eigenvalues associated to the single-layer model defined as :

$$\lambda = \max_{\alpha=1,\dots,n} (|u_\alpha + \sqrt{gh}|, |u_\alpha|, |u_\alpha - \sqrt{gh}|)$$

$$\mu = \max_{\alpha=1,\dots,n} (|v_\alpha + \sqrt{gh}|, |v_\alpha|, |v_\alpha - \sqrt{gh}|)$$

CUDA implementation

- Considering each thread as a volume control.
- The code architecture :
  - Compute the source terme R.
  - Compute W^*.
  - Compute the fluxes :
    - The predictor step.
    - The corrector step.
  - Compute W^{n+1}.
- Draw the water height H with OpenGL.
- minimize communication between CPU and GPU

Numerical result

- $\Omega = [-10, 10] \times [-10, 10]$.
- Neumann boundary conditions.
- Initial condition.

$$h(0,x,y) = 1 + \frac{1}{4}\left(1 - \tanh\left(\frac{\sqrt{ax^2 + by^2} - 1}{c}\right)\right)$$

$$u_\alpha(0,x,y) = 0 \qquad \text{for } \alpha = 1, \ldots, n$$

$$v_\alpha(0,x,y) = 0 \qquad \text{for } \alpha = 1, \ldots, n$$

where $a = 52$, $b = 25$ and $c = 0.1$.

- The gravity $g = 1$ and the eddy viscosity $\nu = 0.01$.

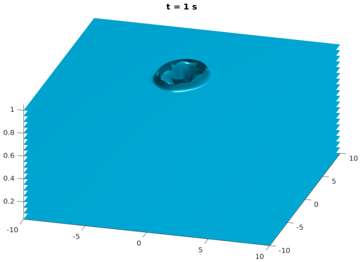
GPU vs CPU

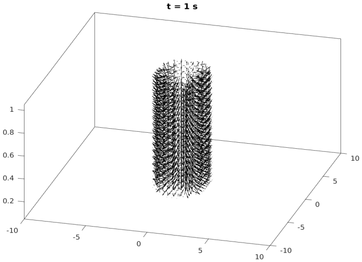
	5 layers			
	50×50	100×100	200×200	500×500
CPU 1 core	0.92	7.62	65.45	1046.14
CPU 8 cores	0.54	3.65	24.76	355.31
GPU	0.47	1.54	6.98	77.26

	10 layers			
	50×50	100×100	200×200	500×500
CPU 1 core	1.81	15.68	134.40	2202.90
CPU 8 cores	0.96	6.70	45.25	738.15
GPU	0.77	2.82	13.29	157.0

	20 layers			
	50×50	100×100	200×200	500×500
CPU 1 core	3.43	31.94	273.74	4349.97
CPU 8 cores	1.87	12.90	96.85	1496.98
GPU	1.42	5.71	28.71	360.50

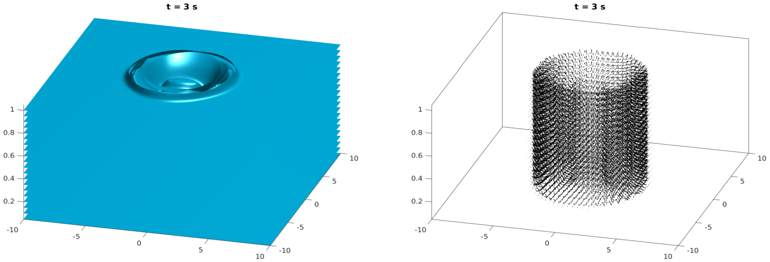
Water heights and velocity fields obtained for $t = 1s$





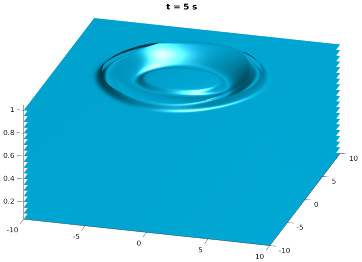
Water heights (left) and velocity fields (right) obtained for dam-break problem with 20 layers for $t = 1s$

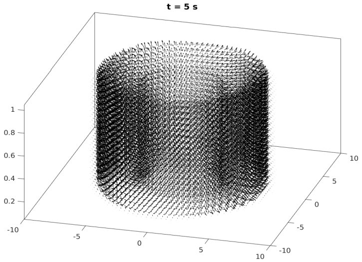
Water heights and velocity fields obtained for $t = 3s$



Water heights (left) and velocity fields (right) obtained for dam-break problem with 20 layers for $t = 3s$

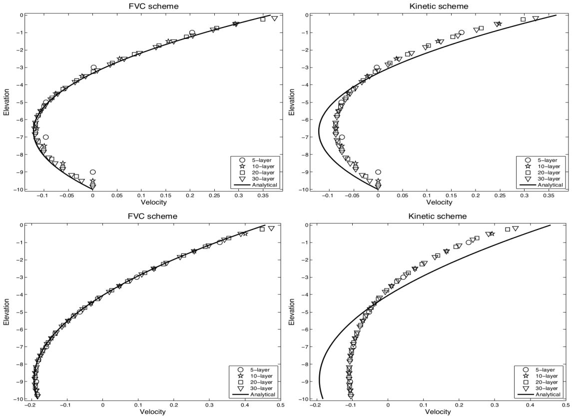
Water heights and velocity fields obtained for $t = 5s$





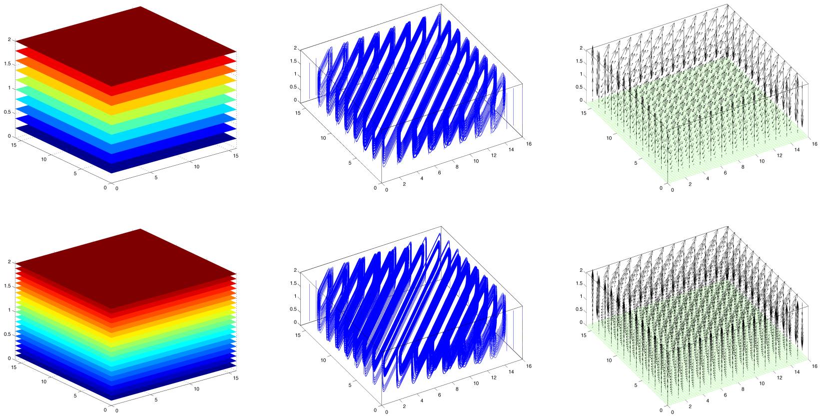
Water heights (left) and velocity fields (right) obtained for dam-break problem with 20 layers for $t = 5s$

Wind-driven circulation flow problem 1D case



Comparisons of numerical predictions with the analytical solution for the wind-driven circulation flow : (top) with bottom friction, (bottom) without bottom friction : (left) FVC scheme, (right) Kinetic scheme.

Wind-driven circulation flow 2D case



Wind circulation on a flat bottom 10 and 20 layers

The multi-layer shallow water model over movable bed

- To update the bed-load in the multilayer system, the Exner equation is used :

$$(1-p)\frac{\partial Z}{\partial t} + \frac{\partial Q_1(u_1, v_1)}{\partial x} + \frac{\partial Q_2(u_1, v_1)}{\partial y} = 0$$

- p is the sediment porosity assumed to be constant.
- Q_1 and Q_2 are the sediment discharge defined by :

$$Q_1(u_1, v_1) = A u_1 (u_1^2 + v_1^2)^{\frac{m-1}{2}}$$
$$Q_2(u_1, v_1) = A v_1 (u_1^2 + v_1^2)^{\frac{m-1}{2}}$$

- An upwind scheme is used to evaluate Z.

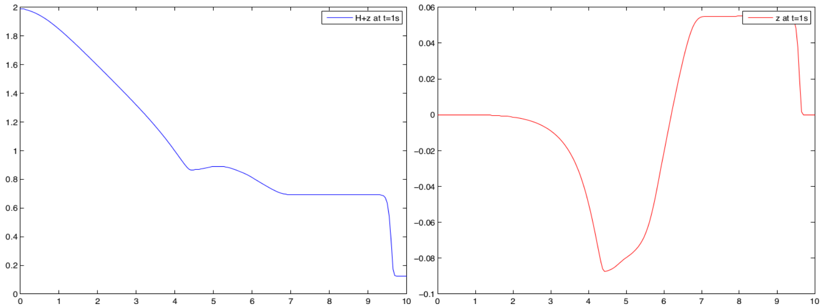
Numerical result

- $\Omega = [0, 10] \times [0, 10]$.
- Neumann boundary conditions.
- Initial condition.

$$h(0, x, y) = \begin{cases} 2 & \text{if } x < 5. \\ 0.125 & \text{elsewhere.} \end{cases}$$

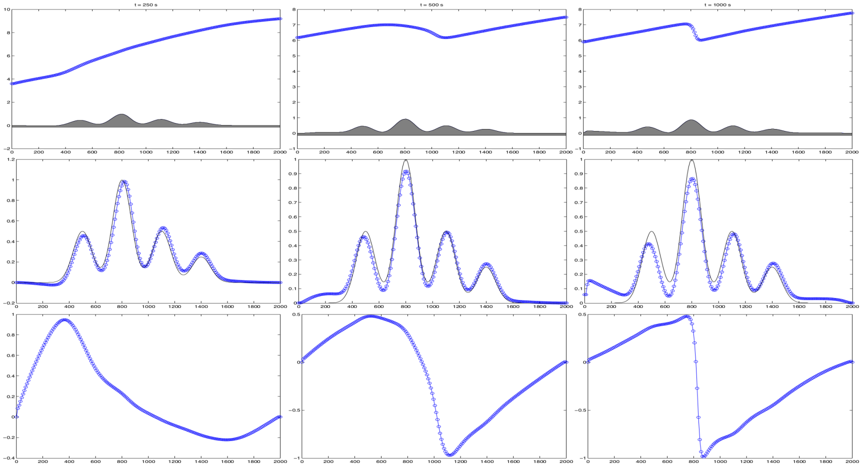
$$u(0, x, y) = 0$$
$$v(0, x, y) = 0$$
$$Z(0, x, y) = 0$$

Dam break problem over erodible bed

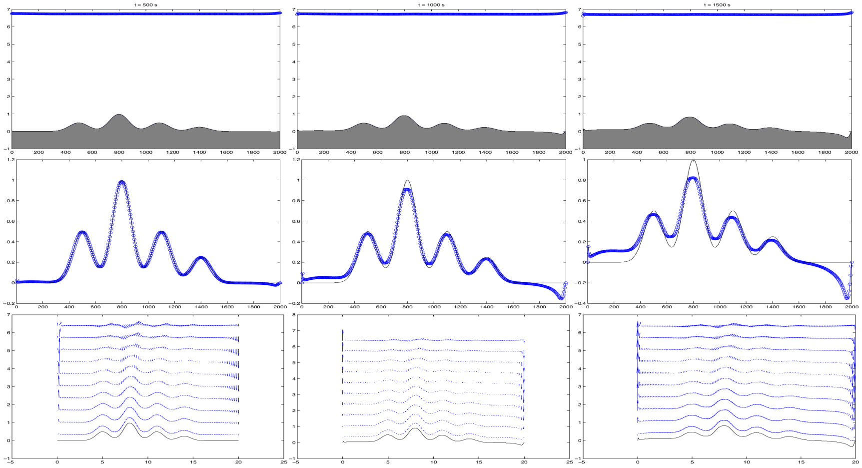


Dam break over a movable bottom - Free surface (left) and Bottom topography (right)

Wind circulation with single layer



Wind circulation with multi layer



Numerical result

- $\Omega = [0, 1000] \times [0, 1000]$.
- Neumann boundary conditions.
- Initial condition.

$$h(0,x,y) = 10 - z(0,x,y)$$

$$u_\alpha(0,x,y) = \frac{Q}{h(0,x,y)}$$

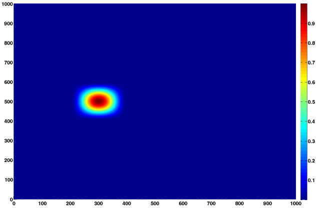
$$v_\alpha(0,x,y) = 0$$

$$Z(0,x,y) = \begin{cases} \sin^2\left(\frac{(x-500)\pi}{200}\right)\sin^2\left(\frac{(y-400)\pi}{200}\right) & \text{if } x \in [500,700]\times[400,600]. \\ 0 & \text{elsewhere.} \end{cases}$$

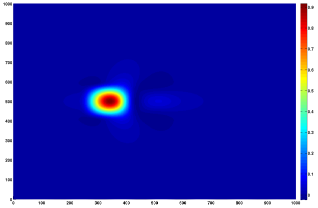
where $Q = 10$, $p = 0.4$ and $A = 1$

- The gravity $g = 9.81$ and the eddy viscosity $\nu = 0.01$.

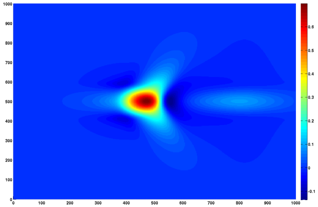
Evolution of the bed (Z)



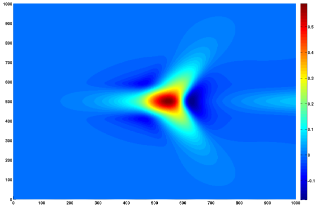
(a) $t = 0$



(b) $t = 200$



(c) $t = 500$



(d) $t = 750$

Conclusions and outlooks

- Conclusions :
  - A GPU implementation of FVC scheme is presented.
  - Solving single and multi-layered shallow water equation in two dimensions.
  - Comparaison with CPU's simulations.
  - Coupling with Exner equation.
- Outlooks :
  - Using Meyer-Peter model instead Grass model.
  - Using unstructured mesh.

Thanks for your attention !

- Boubekeur M., Benkhaldoun F., Schéma FVC : Implémentation sur GPU pour l'approximation des équations de Saint-Venant, TAMTAM 2017, ARIMA.
- Boubekeur M., Benkhaldoun F., Seaid M., GPU accelerated finite volume methods for three-dimensional shallow water flows, FVCA 8, Springer.
- Benkhaldoun F., Seaid M., A simple finite volume method for the shallow water equations, Journal of Computational and Applied Mathematics, vol. 234, no. 1 : (2010) pp. 58–72.
- Audusse E, Benkhaldoun F., Sari S., Seaid M., Tassi P., : A fast finite volume solver for multi-layered shallow water flows with mass exchange, Journal of Computational Physics, vol. 272 : (2014) pp. 23–45.
- Benkhaldoun F., Sari S., Seaid M., Projection finite volume method for shallow water flows, Mathematics and Computers in Simulation, vol. 118 : (2015) pp. 87–101.
- Castro M. J., Ortega S., Asunción M., Mantas J. M., Gallardo J. M., GPU computing for shallow water flow simulation based on finite volume schemes, Comptes Rendus Mécanique, vol. 339, no. 2 : (2011) pp. 165–184.