Adaptive Particle Refinement (APR) applied to SPH method : Stability, accuracy and CPU analysis

Laurent Chiron

LHEEA Lab - Ecole Centrale Nantes

1er juillet 2015

Plan

Plan

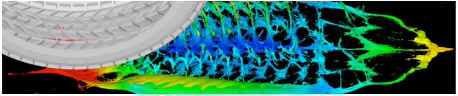
1 SPH background

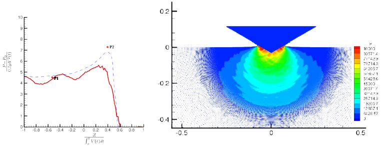
2 Particle Refinement theory

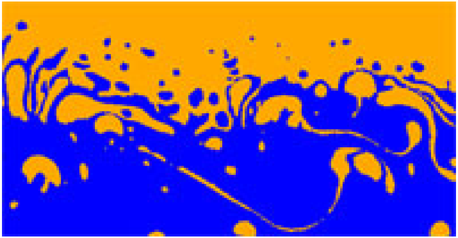
3 Particle refinement improvements
- Arbitrary Lagrangian Eulerian (ALE) process
- The APR concept
- Parallelization

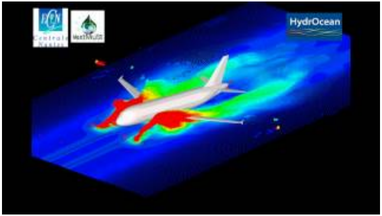
4 Some results
- Kleefsman dambreak

Sph-flow scopes









SPH background

Conservation law

$$\partial_t U + \partial_x f(U) = 0$$

$$\partial_x f(U) \rightarrow \begin{cases} \text{Finite differences} \\ \text{Finite volumes} \\ \text{Discontinuous Galerkin} \\ \text{SPH} \end{cases}$$

SPH background

Conservation law

$\partial_t U + \partial_x f(U) = 0$

$$\partial_x f(U) \rightarrow \begin{cases} \text{Finite differences} \\ \text{Finite volumes} \\ \text{Discontinuous Galerkin} \\ \text{SPH} \end{cases}$$

- How to calculate the term $\partial_x f(U)$ in SPH ?

SPH background

Conservation law

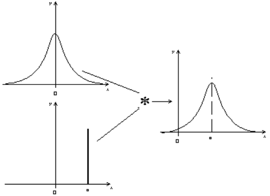
$\partial_t U + \partial_x f(U) = 0$

$$\partial_x f(U) \rightarrow \begin{cases} \text{Finite differences} \\ \text{Finite volumes} \\ \text{Discontinuous Galerkin} \\ \text{SPH} \end{cases}$$

- How to calculate the term $\partial_x f(U)$ in SPH ?

Convolution product properties

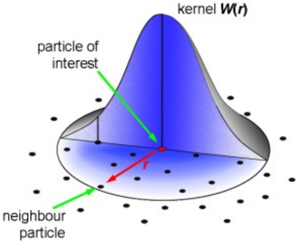
$f * \delta = \int_{\mathbb{R}} f(y)\delta(x-y)dy = f$ $\qquad$ $\nabla f * g = f * \nabla g$



δ is approached by a kernel function W

$$\nabla f_i \approx \nabla f * W = f * \nabla W = \sum_j w_j f_j \nabla W_{ij}$$

SPH background



Benefits

- Meshless (particles)
- Lagrangian method

Weacknesses

- Low spatial order
- No convergence theorem

"Convergence" conditions

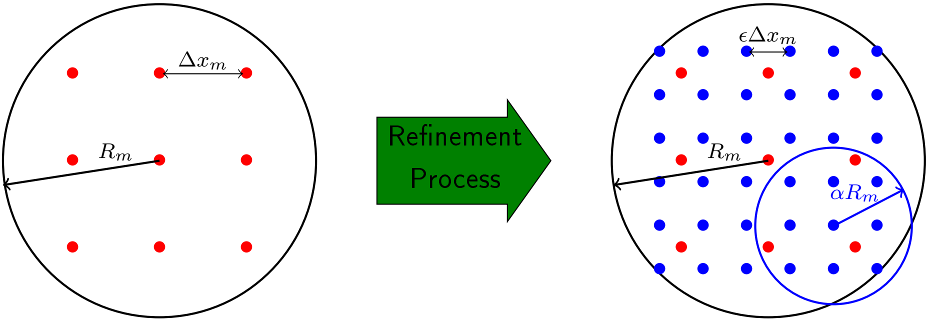
- $\frac{h}{\Delta x} \to \infty$ and $h \to 0$

Plan

Particle refinement theory

Some parameters

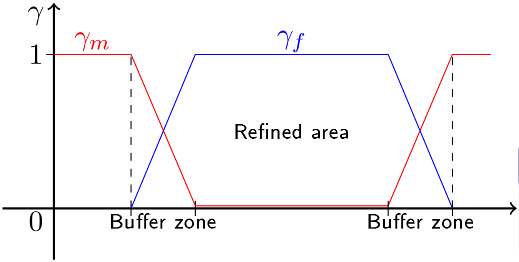
- Spacing parameter $\epsilon \in [0, 1]$
- Radius ratio $\alpha \in]0, 1]$
- Switch parameter $\gamma \in [0, 1]$



Particle refinement theory

Switch parameter properties

- $\gamma = 1 \rightarrow$ particle is turned on (active particle)
- $\gamma = 0 \rightarrow$ particle is turned off (passive particle)



Modified SPH operator

- $\langle \nabla f \rangle_i = \sum_j f_j \omega_j \nabla W_{ij} \gamma_j$

Plan

Plan

Arbitrary Lagrangian Eulerian (ALE) process

$$\frac{d\vec{x}_i}{dt} = \vec{v}_{0i}, \qquad (1)$$

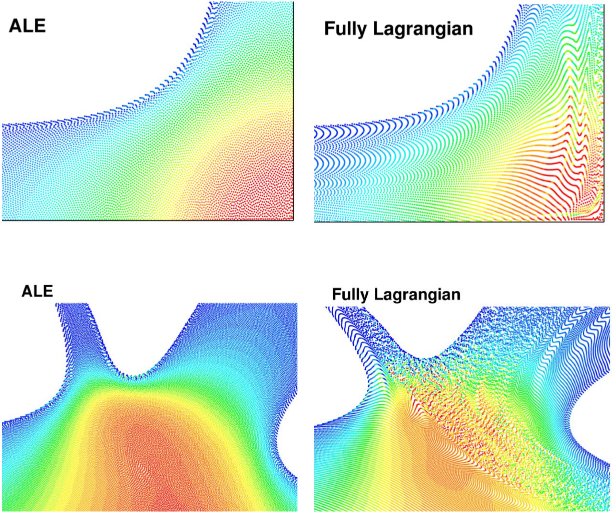
$$\frac{d\omega_i}{dt} = \omega_i \sum_j \omega_j (\vec{v}_{0j} - \vec{v}_{0i}) \nabla W_{ij}, \qquad (2)$$

$$\frac{d\omega_i \rho_i}{dt} = -\omega_i \sum_j \omega_j 2\rho_E (\vec{v}_E - \vec{v}_0(\vec{x}_{ij})) \nabla W_{ij}, \qquad (3)$$

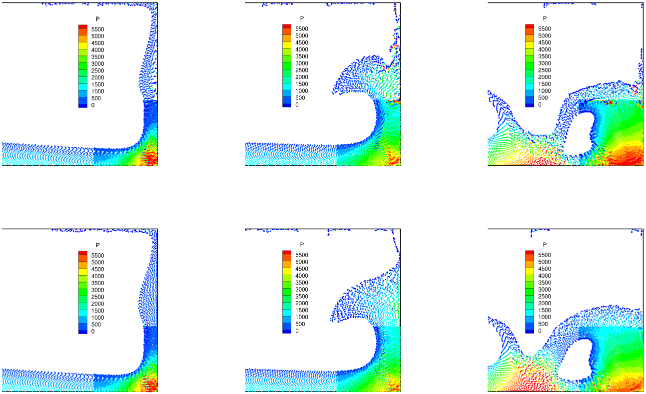
$$\frac{d\omega_i \rho_i \vec{v}_i}{dt} = -\omega_i \sum_j \omega_j 2 \left(\rho_E \vec{v}_E \otimes (\vec{v}_E - \vec{v}_0(\vec{x}_{ij})) + P_E \bar{\bar{I}} \right) \nabla W_{ij} + \omega_i \rho_i \vec{g}, \qquad (4)$$

3 possibilities : $\vec{v}_{0i} = \begin{cases} \vec{v}_i & \text{(Lagrangian)} \\ 0 & \text{(Eulerian)} \\ \vec{v}_i + \vec{\delta}_{vi} & \text{(ALE)} \end{cases}$

ALE process : benefits on a dambreak test case



Why using such an approach ?



Pressure instabilities at the fine/coarse interface

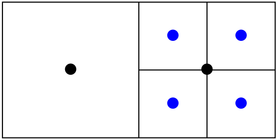
- Particles move in a Lagrangian way (Streamlines)

Plan

The APR concept

Prolongation

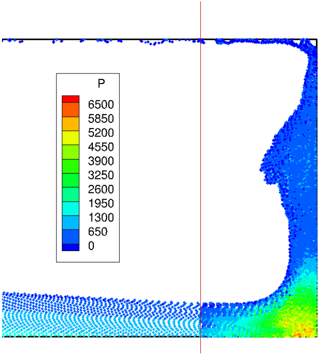
$$\phi_d = \phi_m + {}^R\boldsymbol{D}^h(\phi)_m.(\boldsymbol{x}_d - \boldsymbol{x}_m)$$



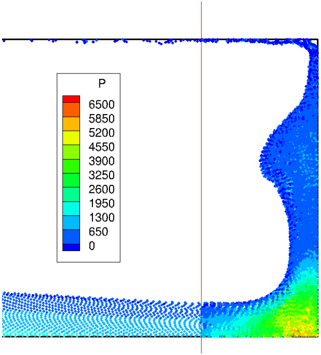
Quantity	Before refinement	After refinement
Mass	m_m	$\sum_j^D m_j$
Kinetic energy	$\frac{1}{2} m_m \vec{u_m}.\vec{u_m}$	$\frac{1}{2}\sum_j^D m_j \vec{u_j}.\vec{u_j}$
Linear momentum	$m_m \vec{u_m}$	$\sum_j^D m_j \vec{u_j}$
Angular momentum	$\vec{x_m} \times m_m \vec{u_m}$	$\sum_j^D \vec{x_j} \times m_j \vec{u_j}$

Prolongation

$$\phi_d = \phi_m + {}^R\boldsymbol{D}^h(\phi)_m.(\boldsymbol{x}_d - \boldsymbol{x}_m)$$



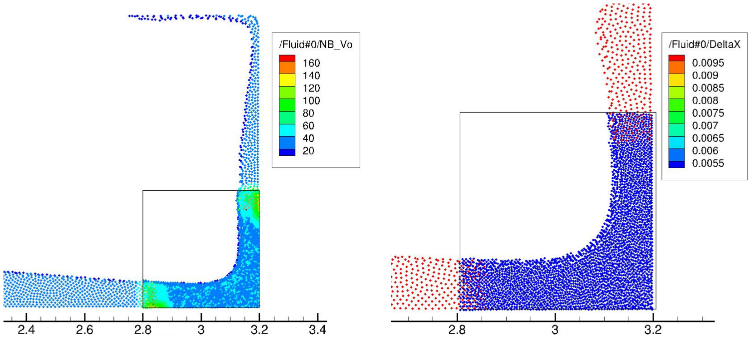
Without prolongation



With prolongation

Plan

The utility of well distributed tasks between processors



- Neighbors number
- Particle radius

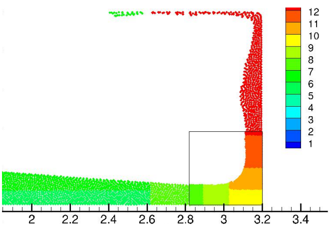
Poor distribution of particles between processors

- Particles inside/near the buffer zone have 2 to 4 more neighbors than other particles

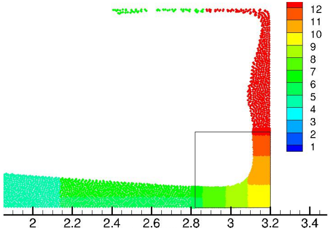
New algorithm efficiency

Load balancing

$\tau = \frac{\min \lambda(k)}{\max \lambda(k)}$ where $\lambda(k)$ is the number of interactions of processor k



- Former algorithm (cutting into an equal number of particles)



- New algorithm (cutting into an equal number of interactions)

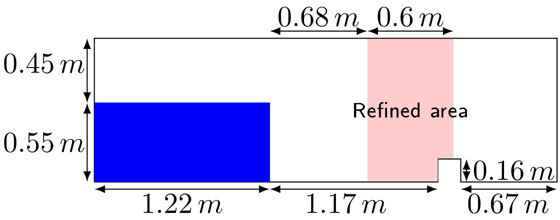
At least of 30% CPU gain

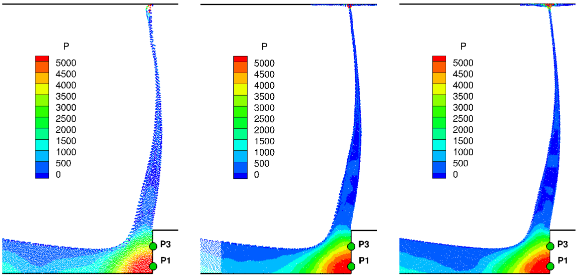
On this test case, $\tau = 0.8$ with the new algorithm (against $\tau = 0.54$ before)

Plan

Plan

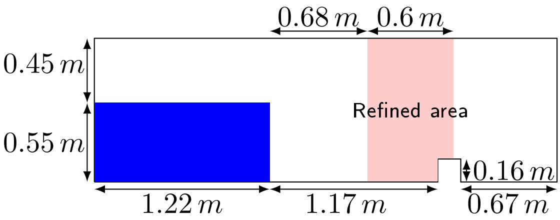
Kleefsman dambreak : pressure field analysis

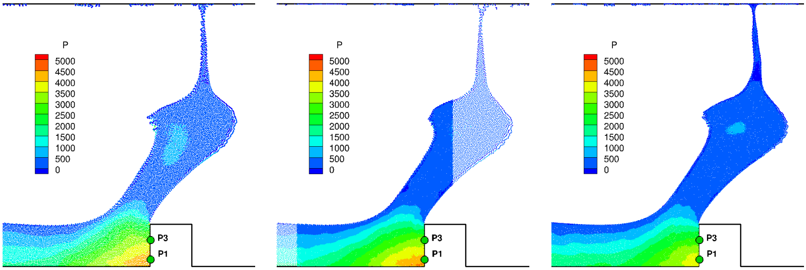




- Coarse
- APR
- Fine

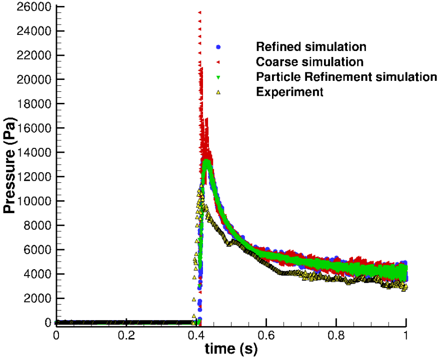
Kleefsman dambreak : pressure field analysis



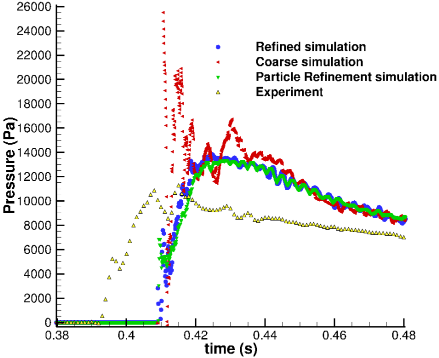


- Coarse
- APR
- Fine

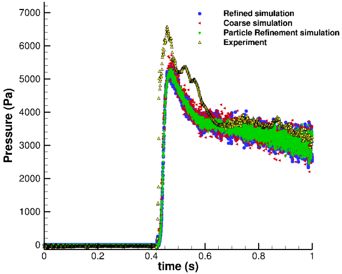
Kleefsman dambreak : Comparison of pressure sensors



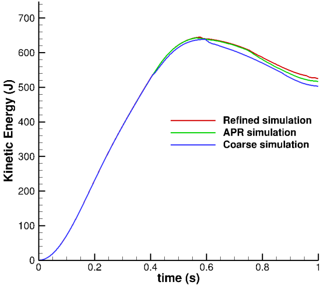
- Sensor P1



- Sensor P1 (Zoom)



- Sensor P3



- Kinetic Energy

Simulation performed	CPU Time (seconds)	Profit
Fine	21 000	-
APR	9000	57.2%

Thank you for your attention !

References

D.A. Barcarolo, G. Oger, and D. Le Touzé.
Adaptive particle refinement and derefinement applied to Smoothed Particle Hydrodynamics method.
Journal of computer physics, 273 :640–657, 2014.

J.Feldman and J.Bonet.
Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems.
Int. J. Numer. Meth. Eng, 72 :295–324, 2007.

S. Kitsionas and A. Whitworth.
Development of SPH variable resolution using dynamic particle coalescing and splitting.
Monthly Notices of the Royal Astronomical Society, 330 :129–136, 2002.

K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, and B. Buchner.
A volume-of-fluid based simulation method for wave impact problems.
Journal of Computational Physics, 206 :363–393, 2005.

Y. R. Lopez, D. Roose, and C. R. Morfa.
Dynamic particle refinement in SPH : application to free surface flow and non-cohesive soil simulations.
Comput Mech, 51 :731–741, 2013.

S. Marrone, A. Colagrossi, D. Le Touzé, and G. Graziani.
Fast free-surface detection and level-set function definition in SPH solvers.
Journal of Computational Physics, 229 :3652–3663, 2010.