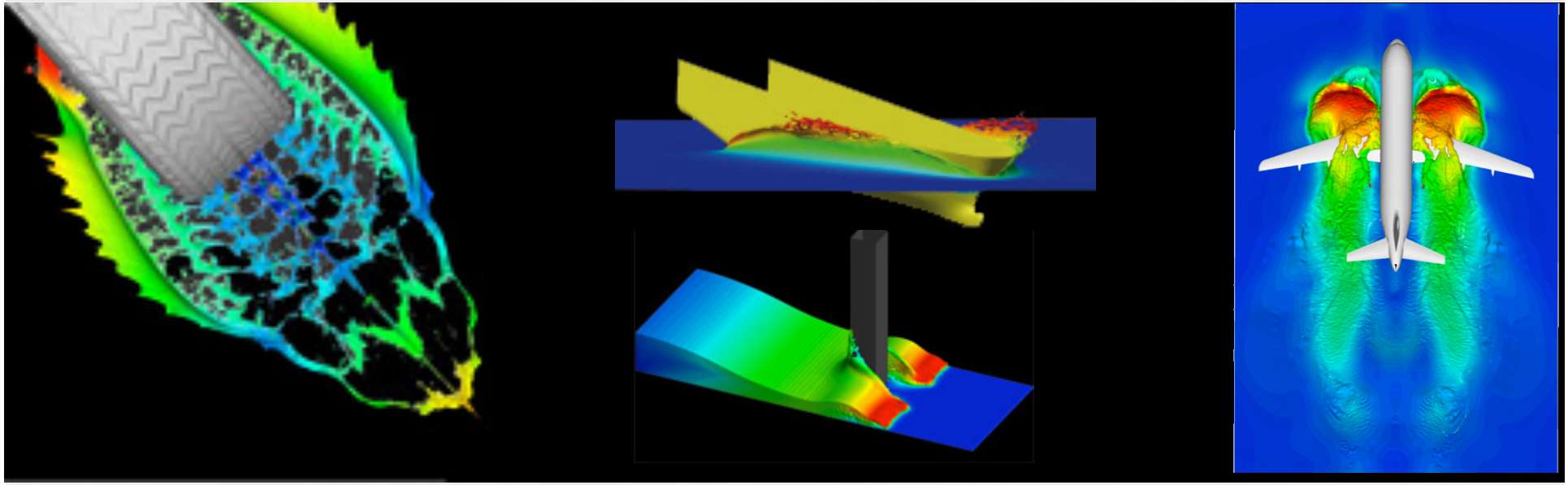
Accélération CPU/GPU d'un code SPH 3D pour des simulations massives en hydrodynamique à surface libre

G. Oger LHEEA (ex LMF), Centrale Nantes / CNRS, Nantes

D. Guibert HydrOcean, Nantes



La collaboration CRIHAN-ECN

2000-2005 : calculs DGA/ECN au CRIHAN, optimisation du code ICARE par le CRIHAN

2005- 2008 : intensification des relations, décision de réaliser un investissement commun

2009 : appel d'offre commun Antares : 1300 cœurs CRIHAN / 280 cœurs ECN

2010-2013 : exploitation commune très efficace

Equipe de recherche DyRaC

Spécialité de l'équipe DyRac: Ecoulements à Dynamique Rapide et Couplages multiphysiques

Physique

- Ecoulements à surface libre à haute dynamique (impacts, fragmentations, effets de compressibilité)
- Interactions violentes entre fluides et structures déformables
- Ecoulements multiphasiques complexes
- MagnetoHydroDynamique

Méthodes

- Méthode Lagrangienne sans maillage (SPH)
- FSI: couplage SPH/FE ou SPH/SPH
- Développement d'un outil CFD basé sur maillage (grille Cartésienne, AMR, explicite)
- Expérimentations avec nos propres moyens d'essais

Reconnaissance

- SPH: code comptant parmi les plus avancés au niveau international
  - Chaire du groupe SPHERIC (ERCOFTAC) : 95 entités académiques et industrielles, 29 pays représentés
  - Leader du projet Européen « NextMuSE » (FP7 Future and Emerging Technologies (7 partenaires, 2.5M €)
  - Publications dans J. Comput. Phys., Phys. Rev. E, Comput. Meth. Appl. Mech. Engng….
- HPC (avec HydrOcean): applications industrielles massives, de l'ordre de 10000 cœurs

Projets

Public (Europe, ANR, DGA) – Dual (IRT, ADEME) – Privé (CITEPH, CIFRE HO, SAIPEM, Michelin)

Smoothed Particle Hydrodynamics: ses intérêts

Spécificité de la méthode SPH:

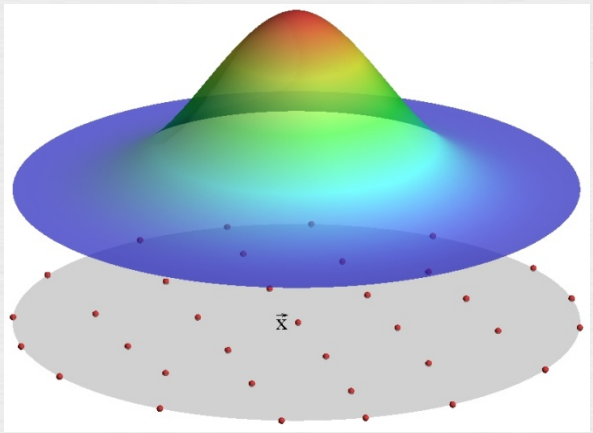
- meshless: absence d'algorithme de gestion de maillage

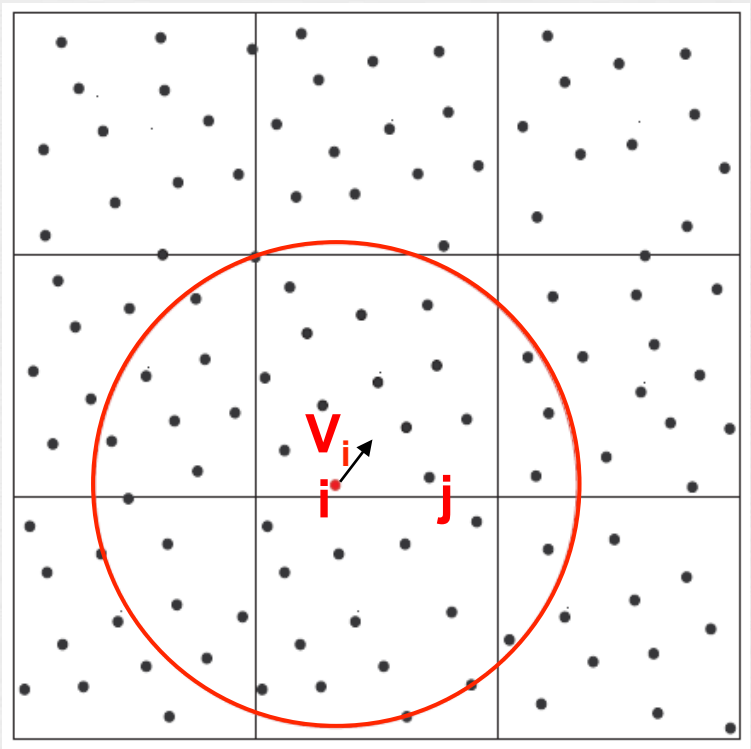
- Lagrangien: pas de discrétisation des termes convectifs
  => précis dans le cadre d'écoulements à dynamique rapide
  (impacts violents, chocs, explosions…)

- Traitement aisé des larges déformations du domaine fluide:
  - Grands déplacements de corps solide(s)
  - Reconnexion/déconnexion d'interface
  (déferlements, jets de surface libre…)

- Interface parfaitement non-diffusive (stricte)

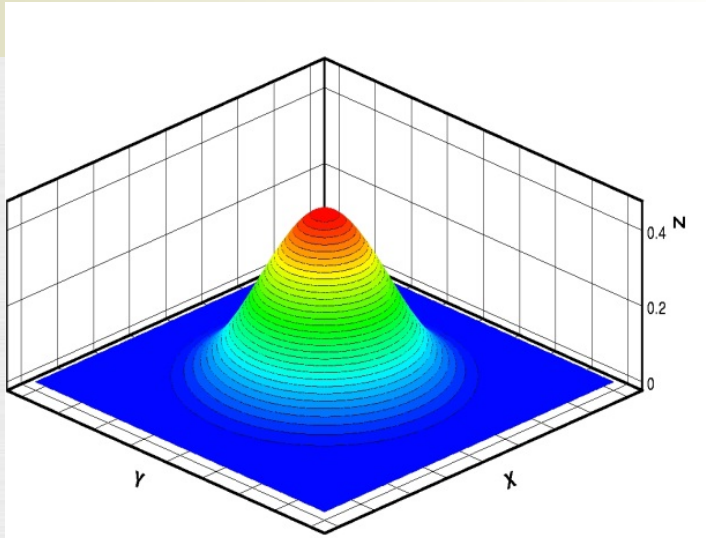
SPH: principe de l'interpolation





- Formalisme Lagrangien
- Absence de connectivités ⇨ équations discrétisées à partir d'un noyau à support compact
- Operateurs différentiels obtenus à partir des valeurs du champ prises aux points inclus dans la zone d'influence

SPH: principe de l'interpolation



Interpolation

convolution à partir d'un noyau W (~ fonction test)

$$f(\vec{r}) \approx \int_D f(\vec{x}) W(\vec{r} - \vec{x}) d\vec{x}$$

$$W(q = \frac{|\vec{r}|}{h}) = C \begin{cases} \frac{2}{3} - q^2 + \frac{1}{2}q^3 & \text{if } 0 \leq q < 1 \\ \frac{1}{6}(2-q)^3 & \text{if } 1 \leq q < 2 \\ 0 & \text{else} \end{cases}$$

Intérêt: détermination du gradient directement à partir des valeurs du champ

$$\nabla f(\vec{r}) \approx \int_D \nabla f(\vec{x}) W(\vec{r} - \vec{x}) d\vec{x} = \int_D f(\vec{x}) \nabla W(\vec{r} - \vec{x}) d\vec{x}$$

analytical

Schéma numérique

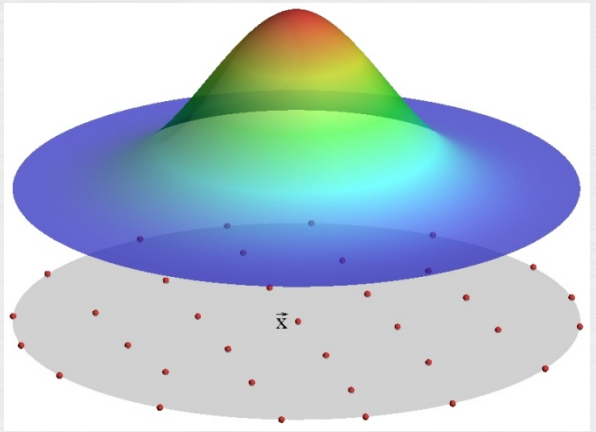
Schéma numérique

- Equations d'Euler $\frac{D\vec{v}}{Dt} = \vec{g} - \frac{\nabla p}{\rho}$
- Fluide compressible $\frac{D\rho}{Dt} = -\rho div(\vec{v})$ $\quad P - P_0 = \frac{\rho_0 c_0^2}{7}\left[\left(\frac{\rho}{\rho_0}\right)^7 - 1\right]$
- Conditions de SL $p(\vec{x},t) = 0$ et $\frac{D\vec{x}}{Dt} = \vec{v}_{\partial\Omega}$ $\forall \vec{x} \in \partial\Omega$ => intrinsèques

  Colagrossi, Antuono, Le Touzé, *Phys. Rev. E*, 2009
- Conditions Initiales $\vec{v}(\vec{x},t_0) = \vec{v}_0$ et $p(\vec{x},t_0) = p_0$

$$\langle \nabla p_i \rangle = \sum_j (p_i + p_j)\omega_j \nabla W(\vec{x}_i - \vec{x}_j)$$

$$\langle div(\vec{v}_i) \rangle = \sum_j (\vec{v}_j - \vec{v}_i)\omega_j \nabla W(\vec{x}_i - \vec{x}_j)$$



Code SPH-Flow : caractéristiques

SPH-flow

Cœur du modèle

- 3D
- Différentes variantes: faiblement-compressible, compressible, incompressible
- ALE
- Differents modèles compressibles (solveur de Riemann, δ-SPH, artificial viscosity)
- Corrections pour montée en ordre : 0th-order (Shepard), 1st-order (Libersky/MLS)…
- Prise en compte de géométries 3D complexes (Ghost particles, Normal Flux method)
- Discrétisation variable en espace
- Conditions d'entrée-sortie
- Parallélisé sur mémoire distribuée (MPI)

Extensions du modèle

- Couplage fluide-structure
- Ecoulements multifluides
- Modèle shallow-water

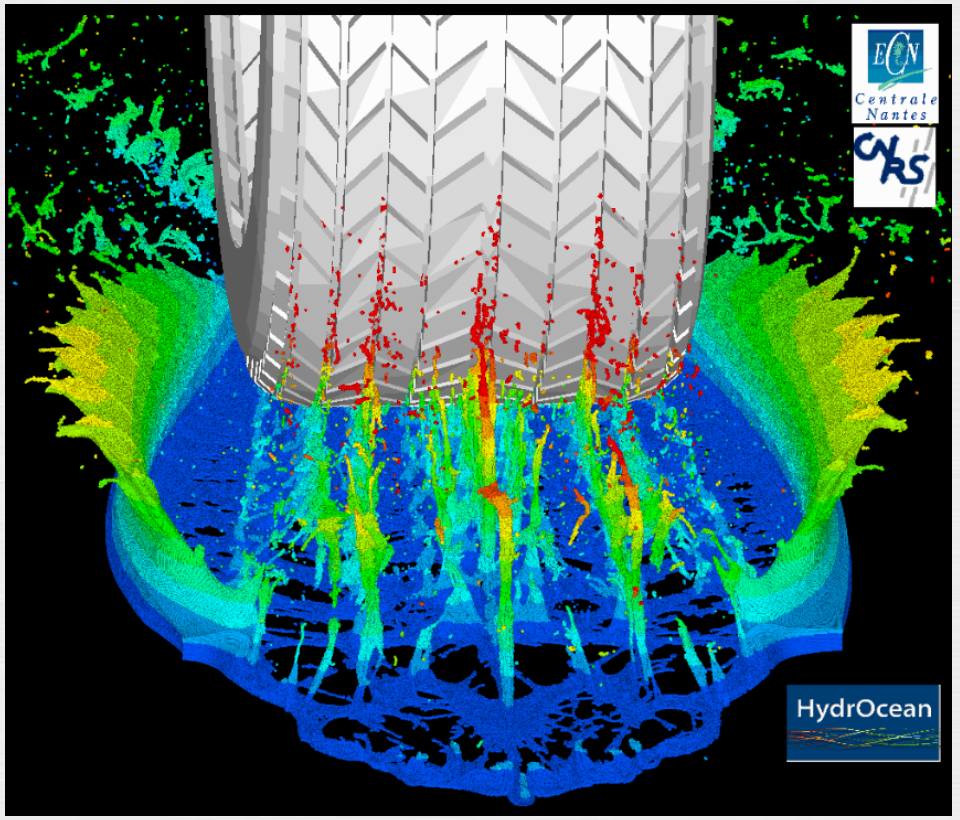
Trois points-clés

- précision
- HPC
- validation

Exemple d'application industrielle

Aquaplanning

- Simulation d'un pneumatique indéformable en roulement sans glissement
- Eau initialement au repos: flaque d'eau d'épaisseur 10 mm
- Vitesse: 130 km/h
- 10 millions de particules
- 32 cœurs

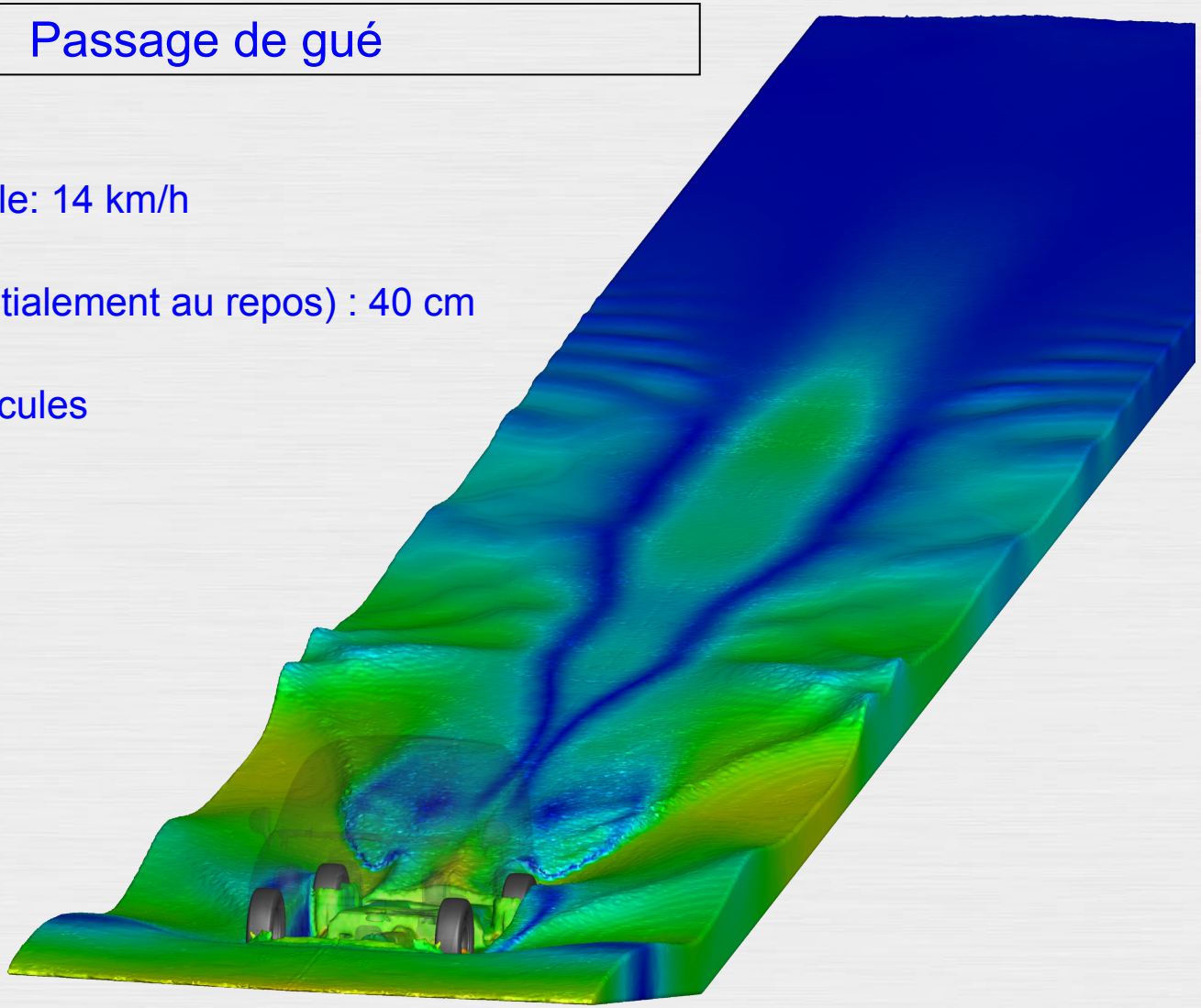


MICHELIN

Exemple d’application industrielle

Passage de gué

- Vitesse du véhicule: 14 km/h
- Hauteur d’eau (initialement au repos) : 40 cm
- 3 millions de particules



Exemple d’application industrielle

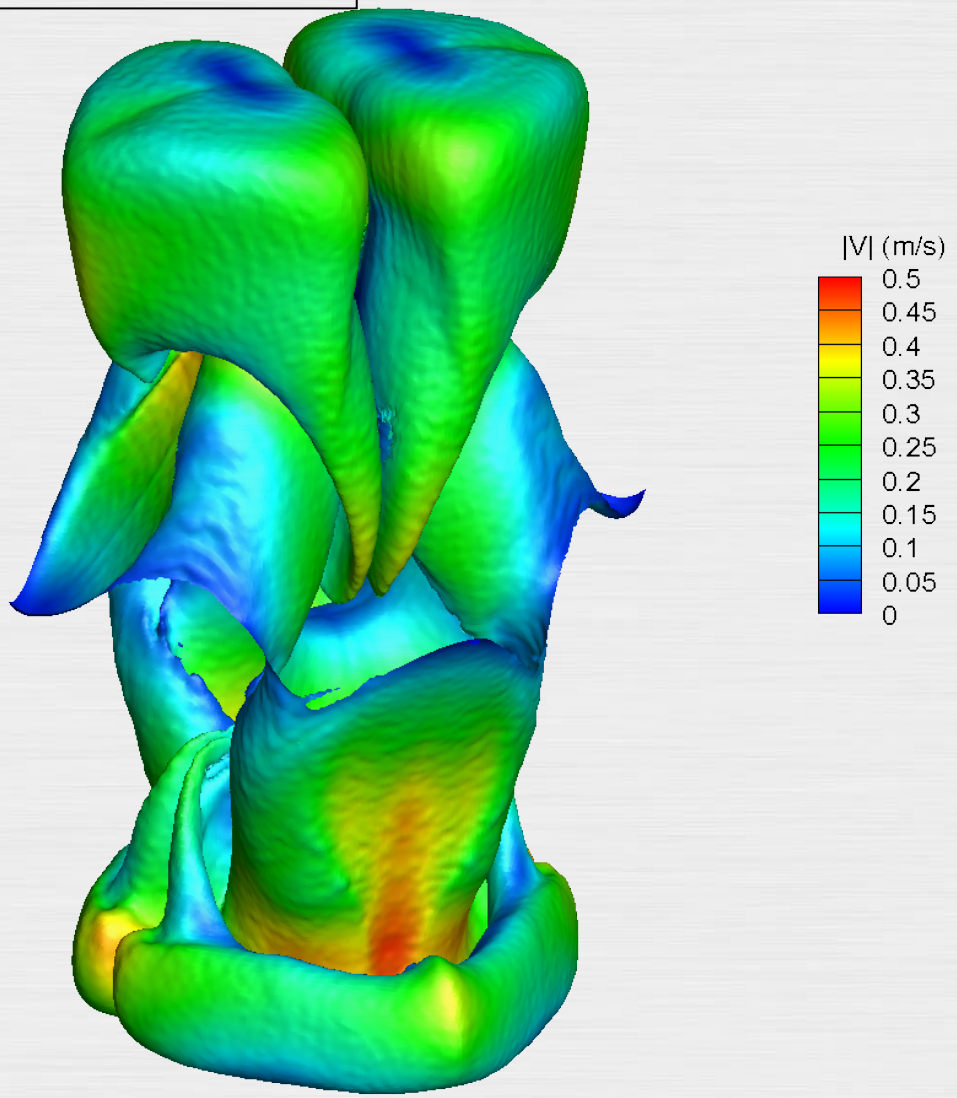
Remontée de bulles multiples

- Eau-huile

Exemple d’application industrielle

Instabilité de Rayleigh-Taylor 3D

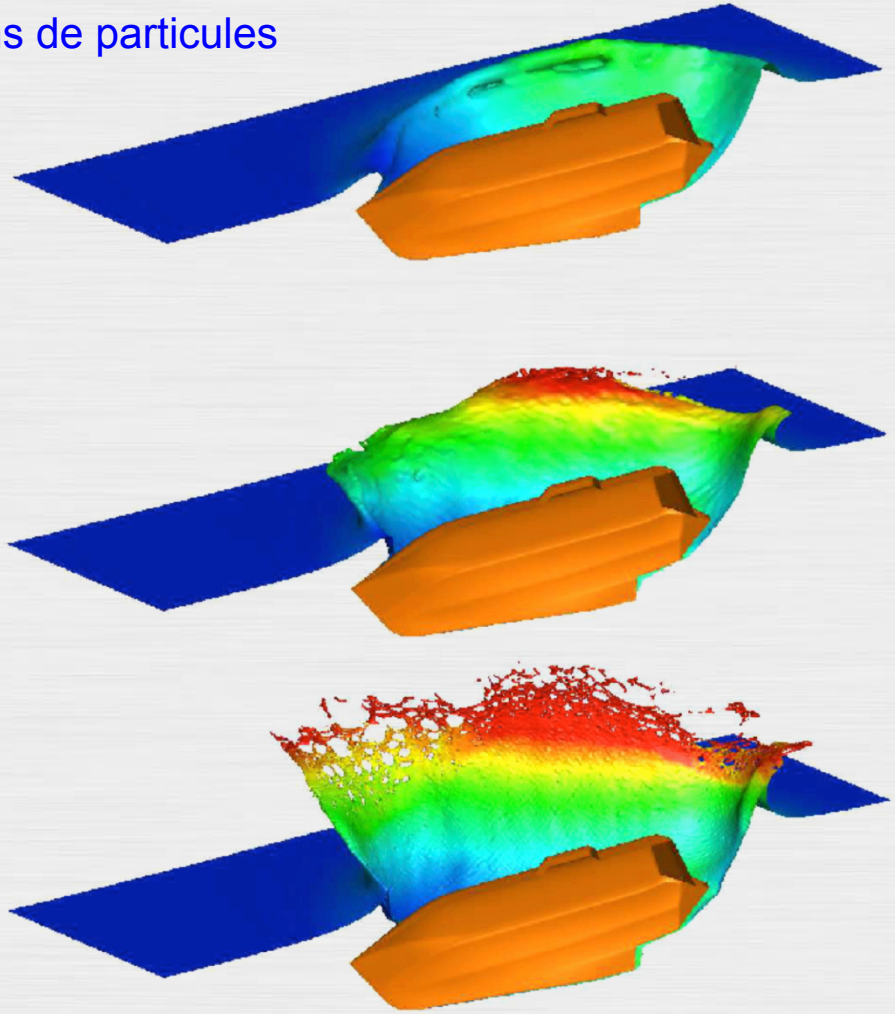
- 6,75 millions de particules
- 64 cœurs
- Durée du calcul: 71650 secondes



Exemple d'application industrielle

Chute libre d'un lifeboat

- Trois discrétisations: 1, 10 et 100 millions de particules
- 128, 256 et 1024 cœurs respectivement



Couplage Fluide-Structure

modèle structure : méthode Eléments Finis

- Equations: $div\sigma + \vec{f}_v = \rho \ddot{\vec{r}}(M,t)$

$$[M][\ddot{q}(t)] + [K][q(t)] = [F(t)]$$

- Intégration temporelle :

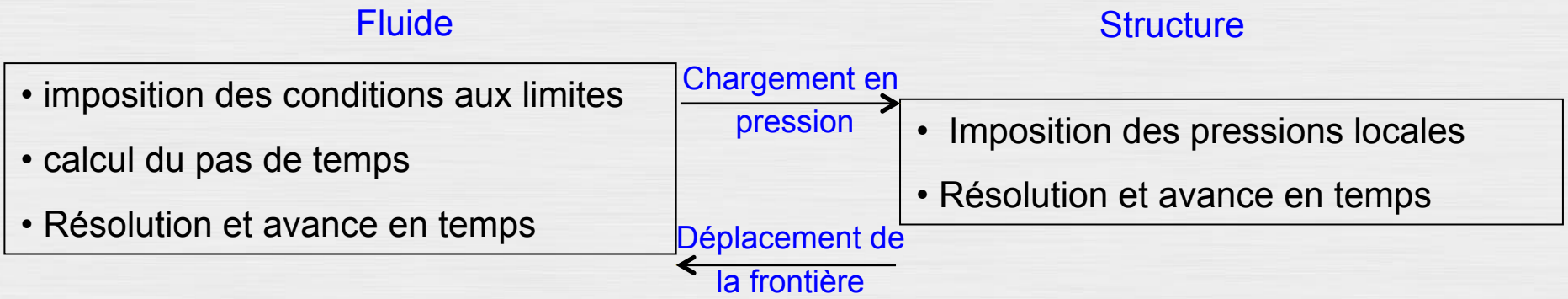
méthode de Hilbert-Hughes-Taylor (HHT) basée sur un schéma de Newmark implicite ⟹ dissipation controllée des oscillations hautes fréquences

Code FEM : *Code_Aster (www.code-aster.org)*

Couplage Fluide-Structure

couplage SPH-FEM

- Couplage faible



- Les pas de temps sont imposés par SPH
- En pratique, plusieurs processeurs sont utilisés pour la partie fluide (SPH) tandis qu'un seul processeur assure le calcul Eléments Finis
- Le calcul Eléments Finis est généralement très rapide, et son temps de restitution est masqué par celui de SPH => scalabilité de SPH-Flow non-affectée par le couplage

Couplage Fluid-Structure

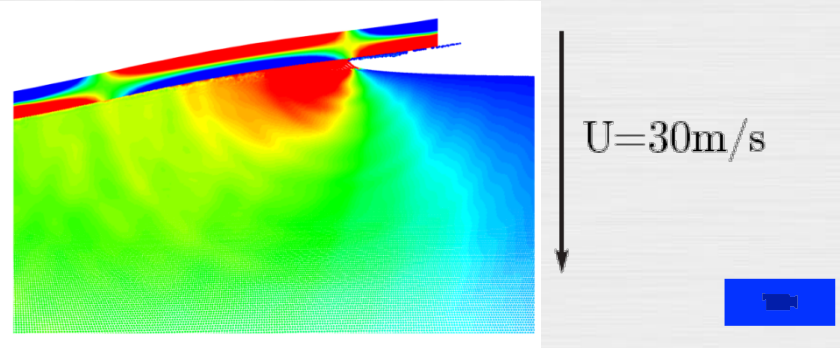
Intérêts du couplage SPH-FEM

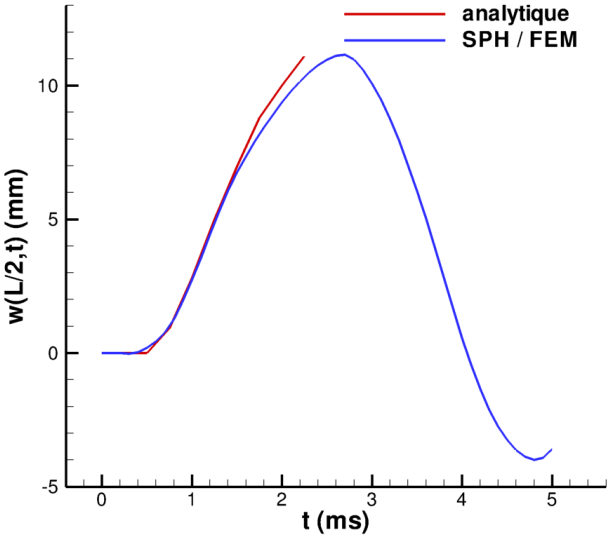
- Les Eléments Finis sont rapides et précis pour simuler les structures
- La méthode SPH traite aisément les fragmentations et reconnexions à la surface libre
- Pas besoin de méthode de suivi ou de capture de surface libre
- Pas besoin de traitement spécifique à l'interface fluide-structure

Cas test de validation

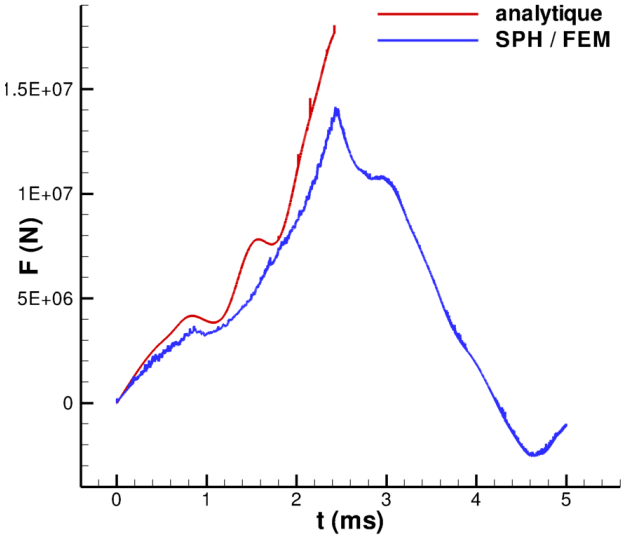
Impact hydro-élastique d'une poutre à la surface libre

- poutre en Aluminium
- haute vitesse d'impact
- poutre encastrée à ses extrémités (imposant la vitesse d'impact)

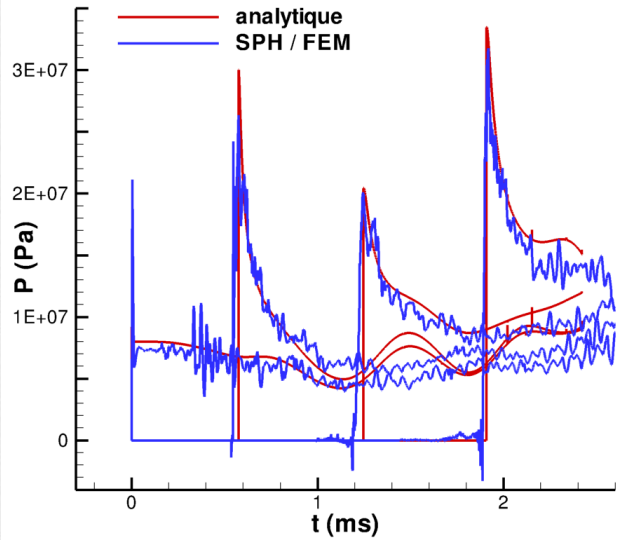




Déformation en milieu de poutre



Force verticale

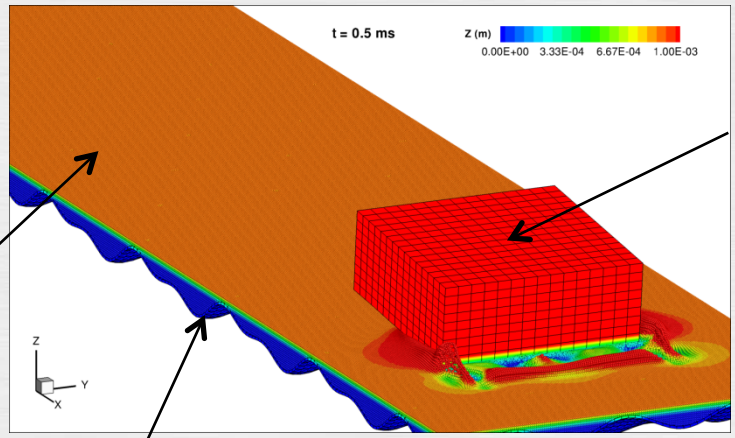


Pression locale

Exemple d'application industrielle

SPH-flow

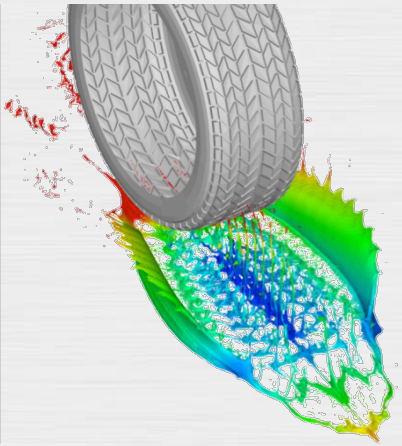
Déformation d'un pneu pendant son passage dans une flaque d'eau initialement au repos

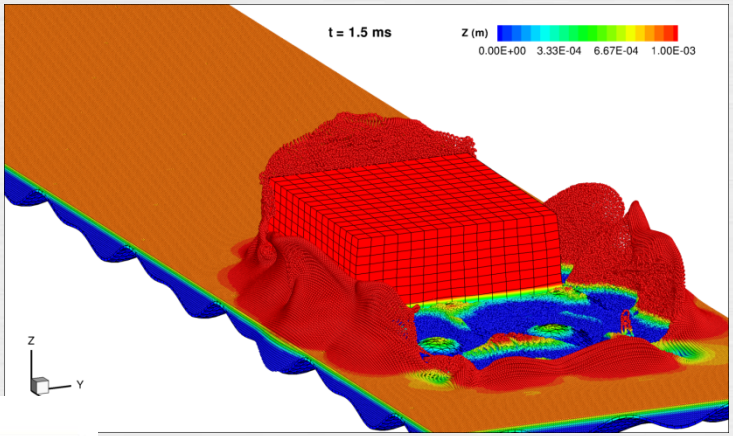


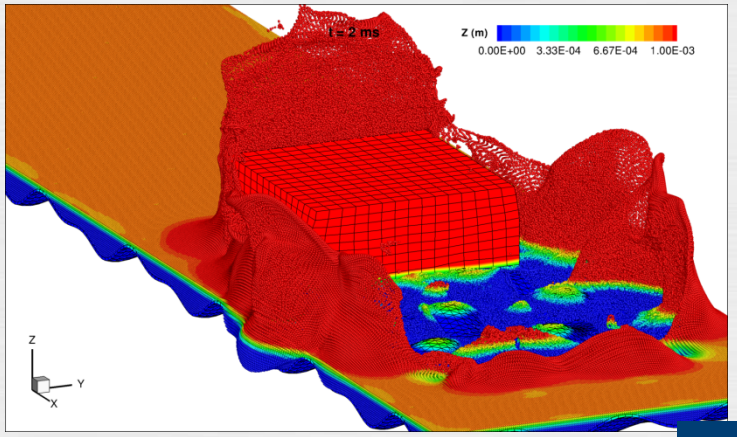
still water

rough road shape

tyre part







Centrale Nantes

cnrs

MICHELIN

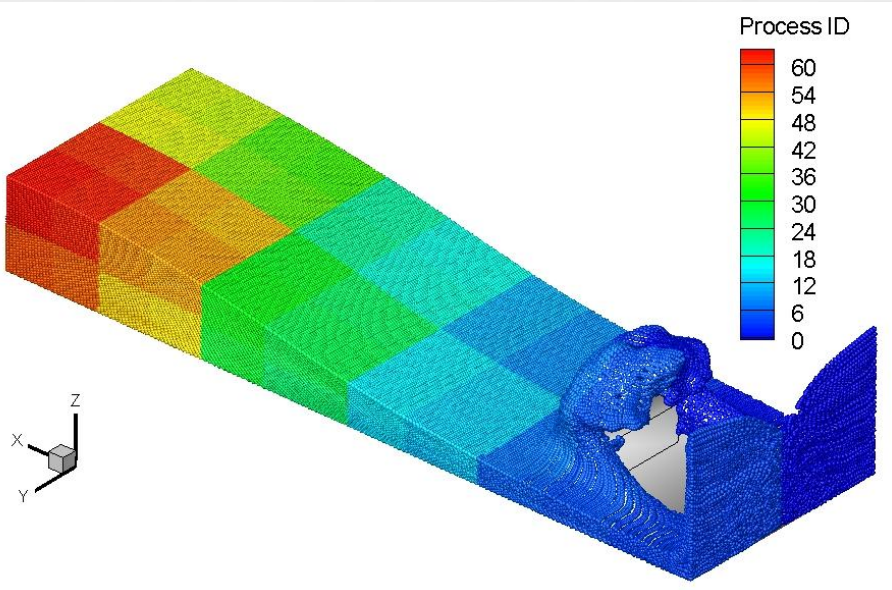
HydrOcean

Parallélisation massive sur CPU en mémoire distribuée

Pourquoi paralléliser?

- L'interpolation par noyau rend la méthode coûteuse
  (en 3D, chaque particule est en interaction avec au minimum 80 particules voisines)
- Le caractère pseudo-compressible de la méthode implique des pas temps très petits
  (pas de temps basés sur une CFL acoustique)
- Le caractère pleinement explicite de la méthode SPH se prête particulièrement à sa parallélisation

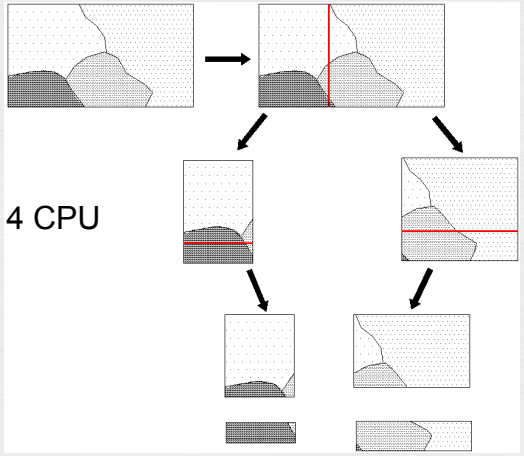
Parallélisation sur mémoire distribuée (MPI)
(Stratégie de décomposition de domaines)



Parallélisation massive sur CPU en mémoire distribuée

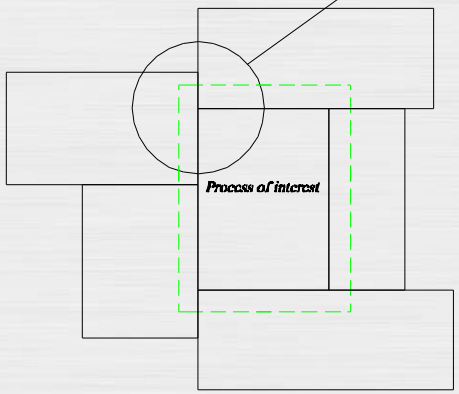
- Domaine fluide total découpé en autant de sous-domaines que de CPU disponibles

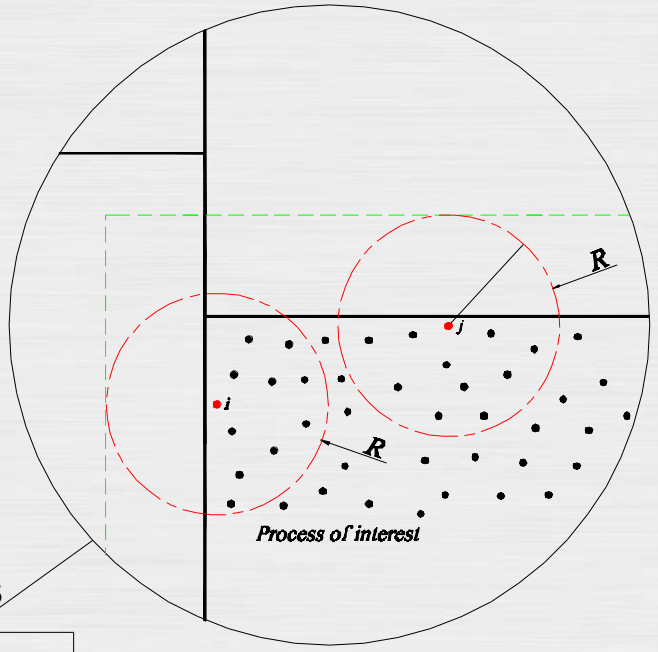
Exemple pour 4 CPU



- Données échangées entre les sous-domaines par communications MPI

- Le support compact du noyau d'interpolation définit les données à échanger





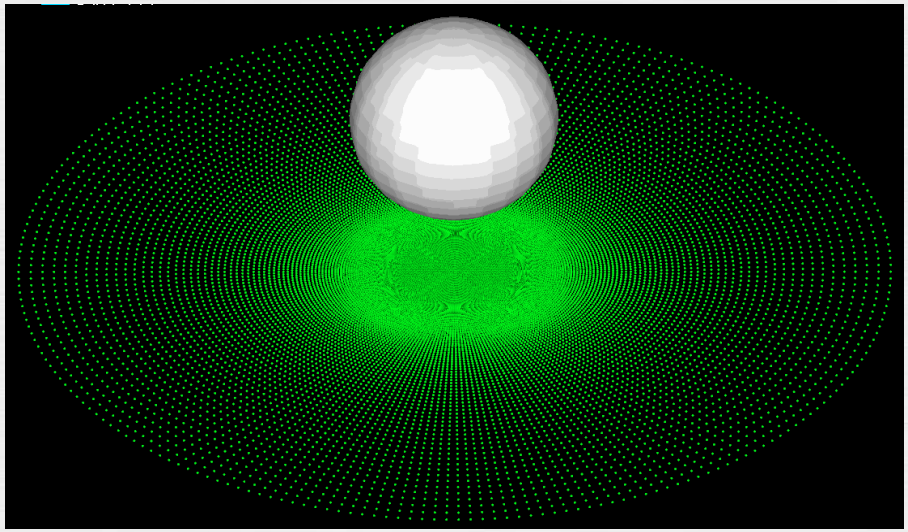
Décomposition de domaine: Load-balance

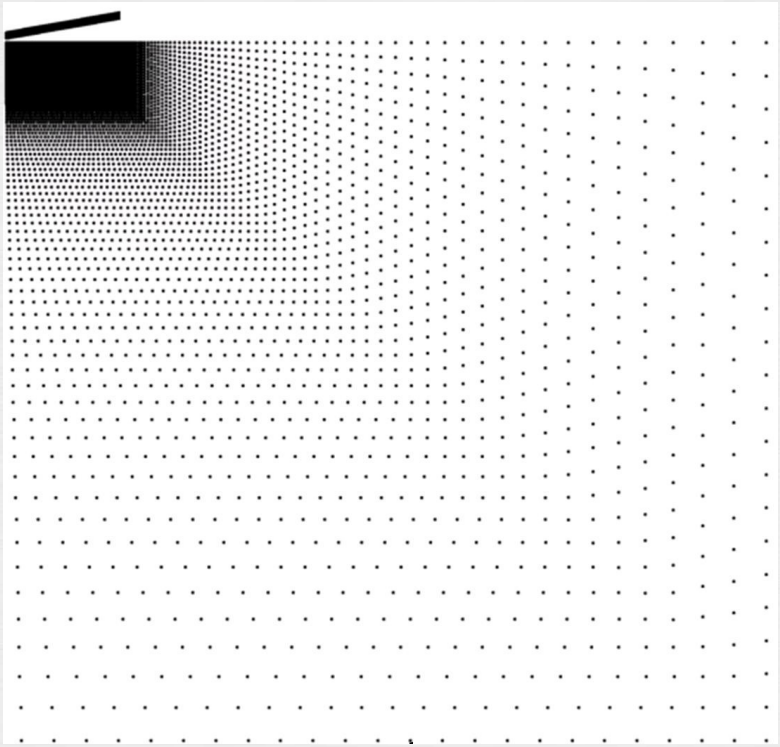
Principe: décomposer *N* particules en *p* sous-domaines

Sous 2 contraintes:

- les particules se déplacent de façon Lagrangienne, donc une mise à jour régulière est nécessaire ⟹ cette procédure doit être performante et scalable !!

- la résolution spatiale peut être variable (R variable en espace)





Décomposition de domaine: Load-balance

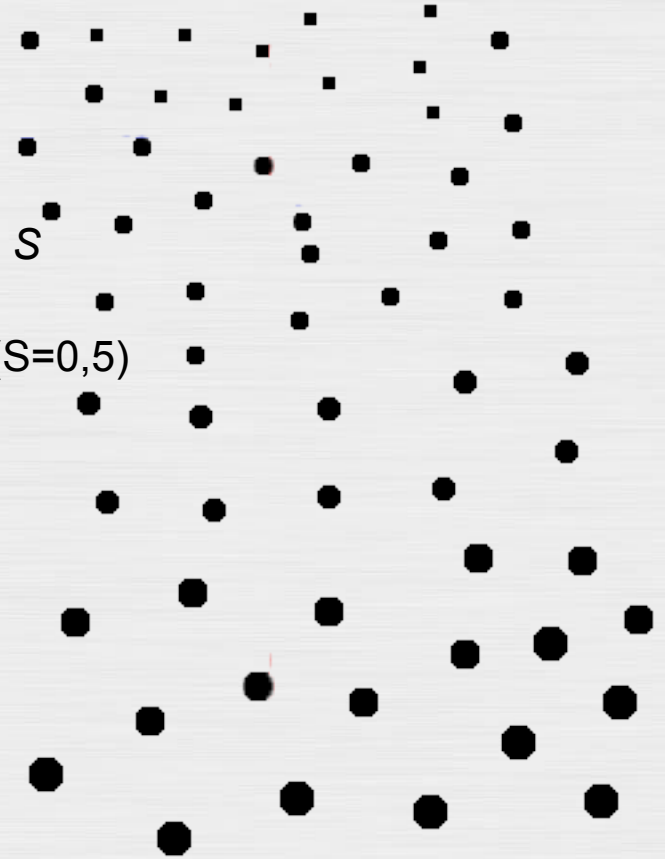
Méthode retenue:

ORB (Orthogonal Recursive Bisection) par échantillonnage

1) Choix du meilleur axe de découpe

2) Construction de la distribution de densité échantillonnée *S*

3) Trouver par dichotomie la position *m* d'équi-distribution (S=0,5)



Décomposition de domaine: Load-balance

Méthode retenue:

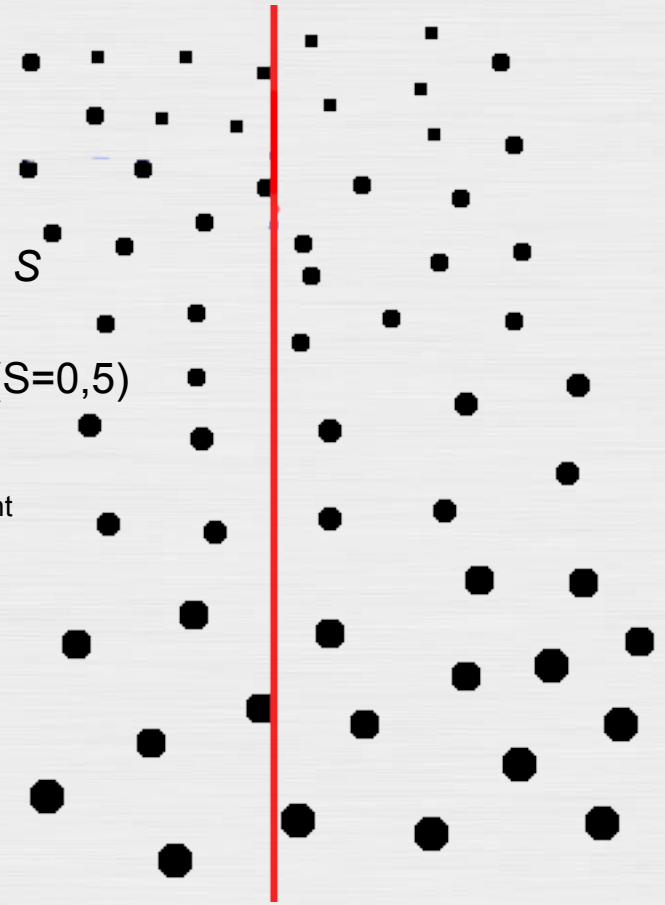
ORB (Orthogonal Recursive Bisection) par échantillonnage

1) Choix du meilleur axe de découpe

2) Construction de la distribution de densité échantillonnée *S*

3) Trouver par dichotomie la position *m* d'équi-distribution (S=0,5)

4) Découpe du domaine D en 2 sous-domaines D_{left} et D_{right}



Décomposition de domaine: Load-balance

Méthode retenue:

ORB (Orthogonal Recursive Bisection) par échantillonnage

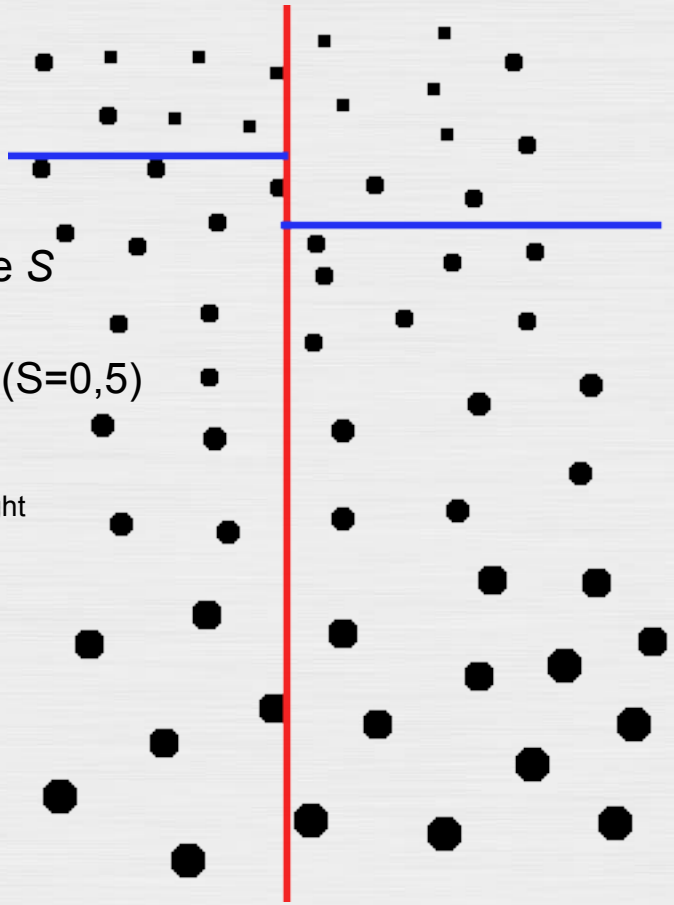
1) Choix du meilleur axe de découpe

2) Construction de la distribution de densité échantillonnée *S*

3) Trouver par dichotomie la position *m* d'équi-distribution (S=0,5)

4) Découpe du domaine D en 2 sous-domaines D_{left} et D_{right}

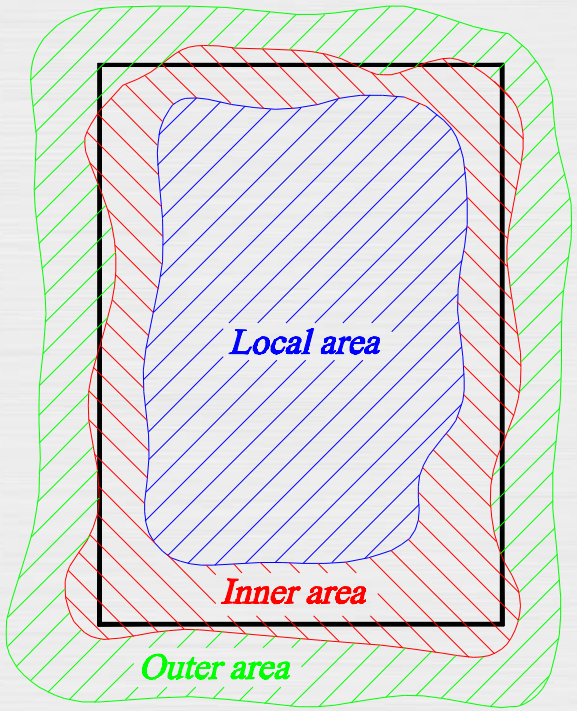
5) Appel de cet algorithme pour D_{left} et D_{right} (récursivité)



Number of cores	4,096	8,192	16,384	32,768
ORB using sampled particles	4.5 s	10.8 s	15.6 s	41.1 s

Temps CPU de l'ORB pour 1 milliard de particules

Masquage des temps de communications au cours du calcul

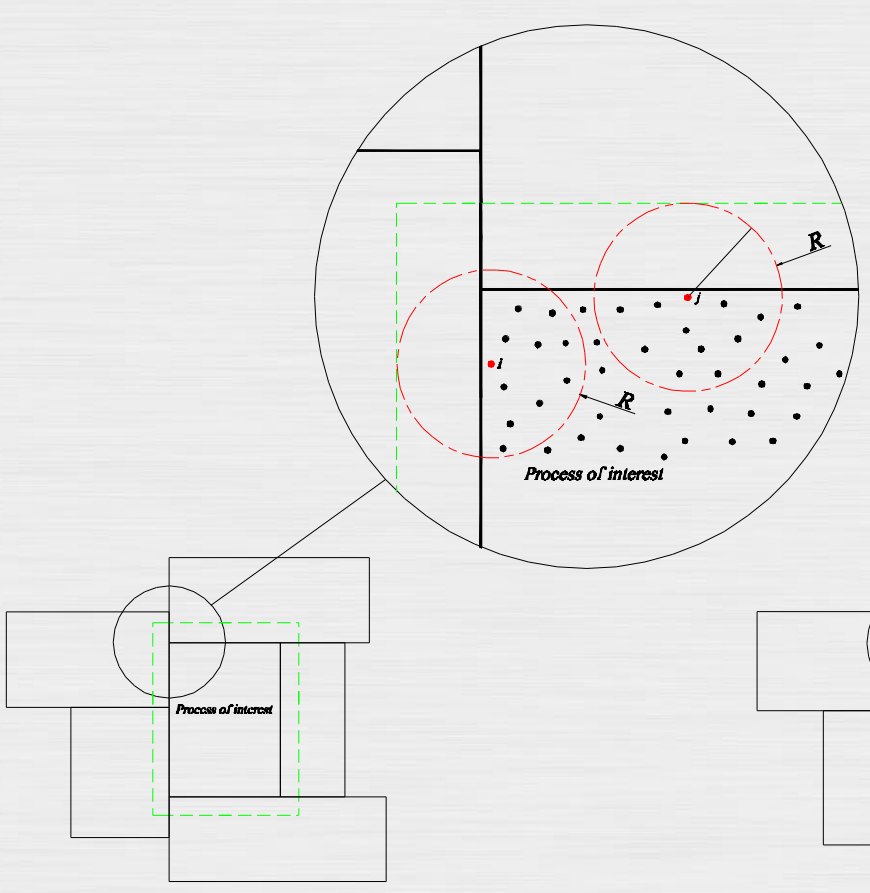


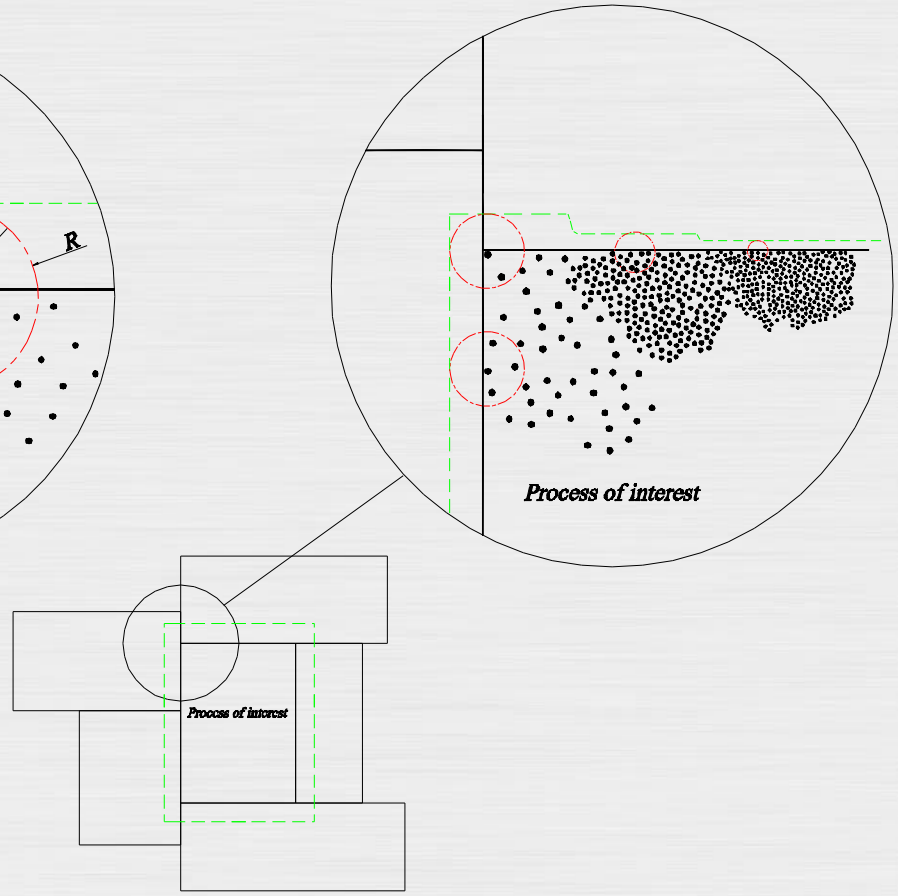
- Introduction d'un système Local-Inner-Outer
- Les « Inner » sont les particules envoyées par le processeur courant à ces voisins
- Les « Outer » sont les particules reçues par le processeur courant de la part de ces voisins
- Les « Local » sont toutes les autres particules
- En principe, le nombre de particules « Local » est a priori très supérieur au nombre de particules « Inner »

⇒ **Principe:** masquer les temps de communication des particules « Inner » par les boucles de calcul des particules « Local »

Difficulté: définition des particules « Inner »

Masquage des temps de communications au cours du calcul



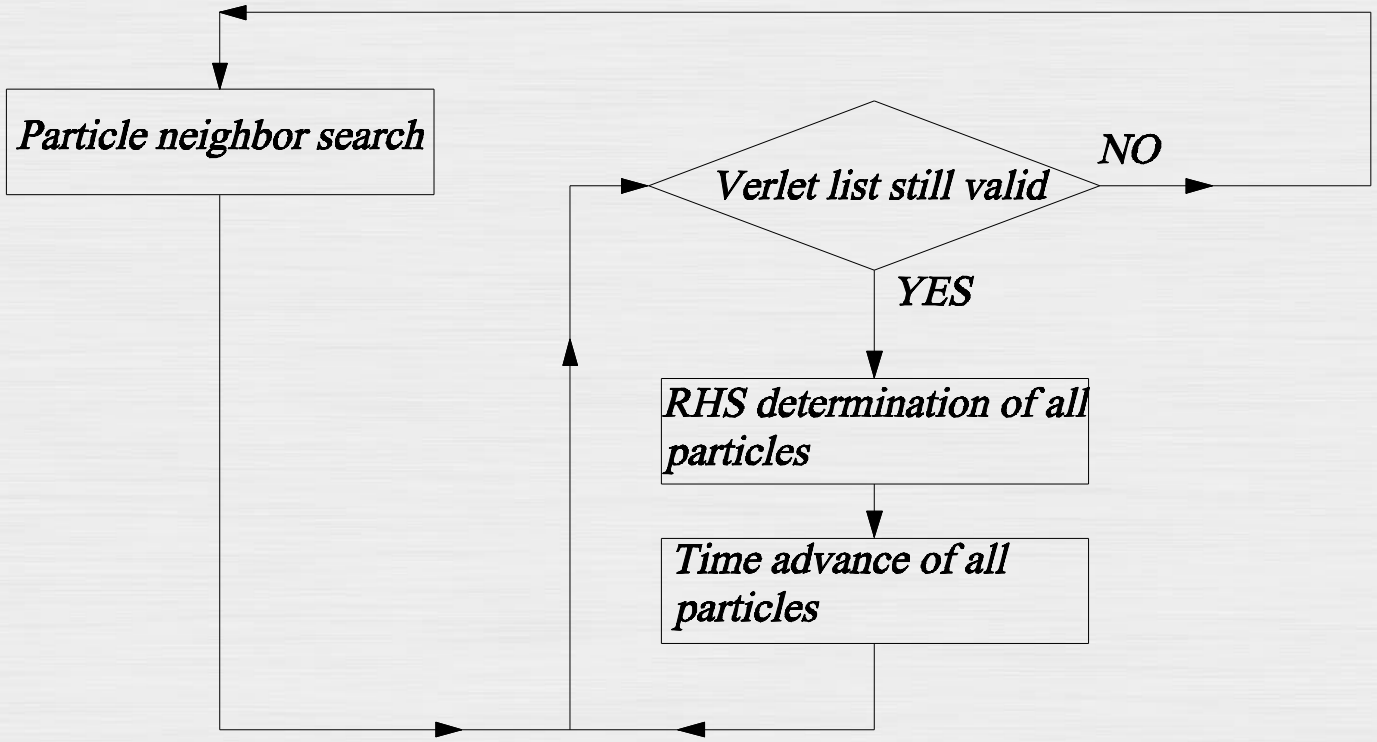


Cas à R constant
(zone « Inner » prismatique ⇨ aisé)

Cas à R variable
(zone « Inner » de forme quelconque !)

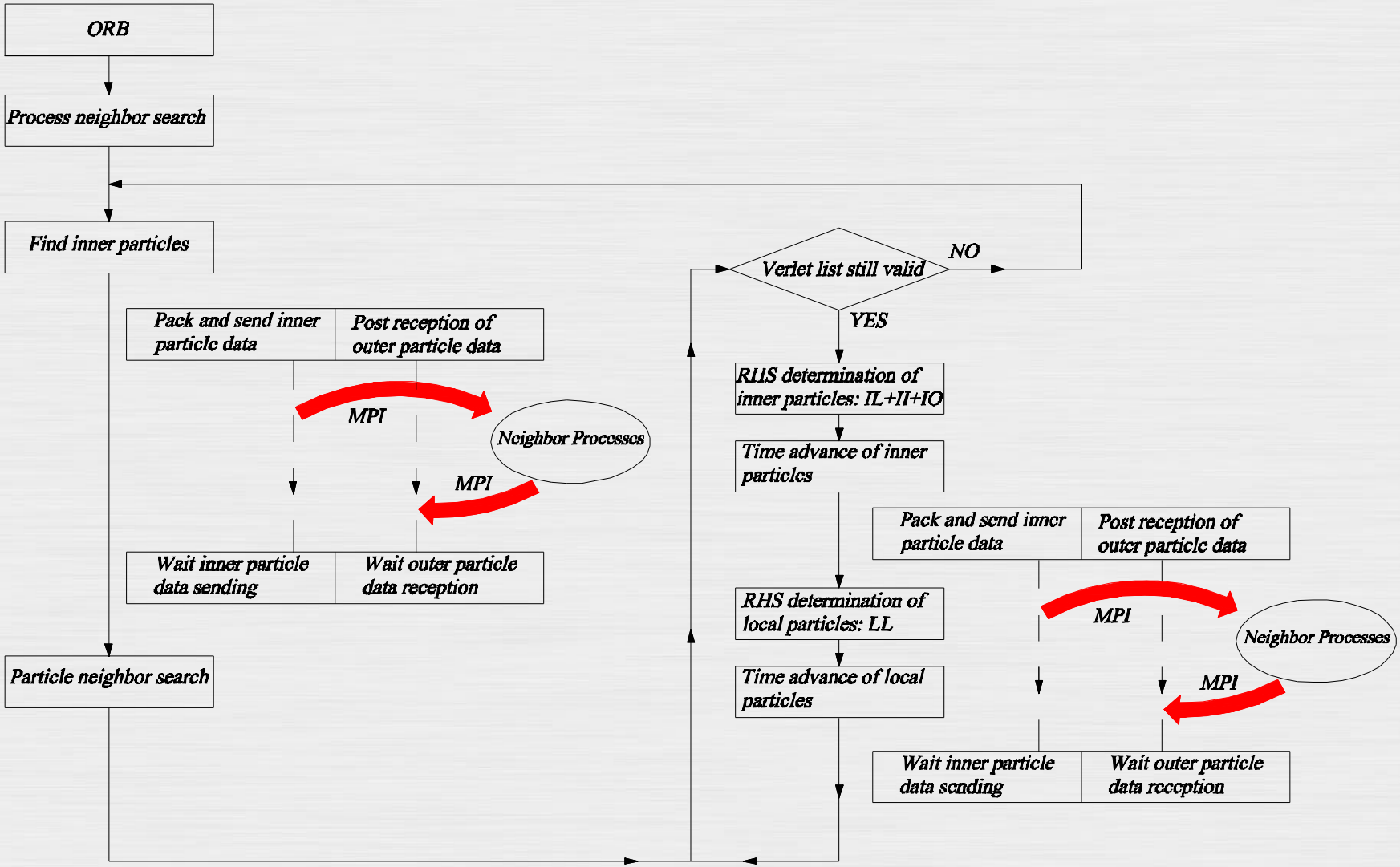
Masquage des temps de communications au cours du calcul

- Diagramme simplifié de fonctionnement séquentiel du code SPH-Flow
  (utilisation de listes de Verlet: recherche non-systématique des voisinages de particules)

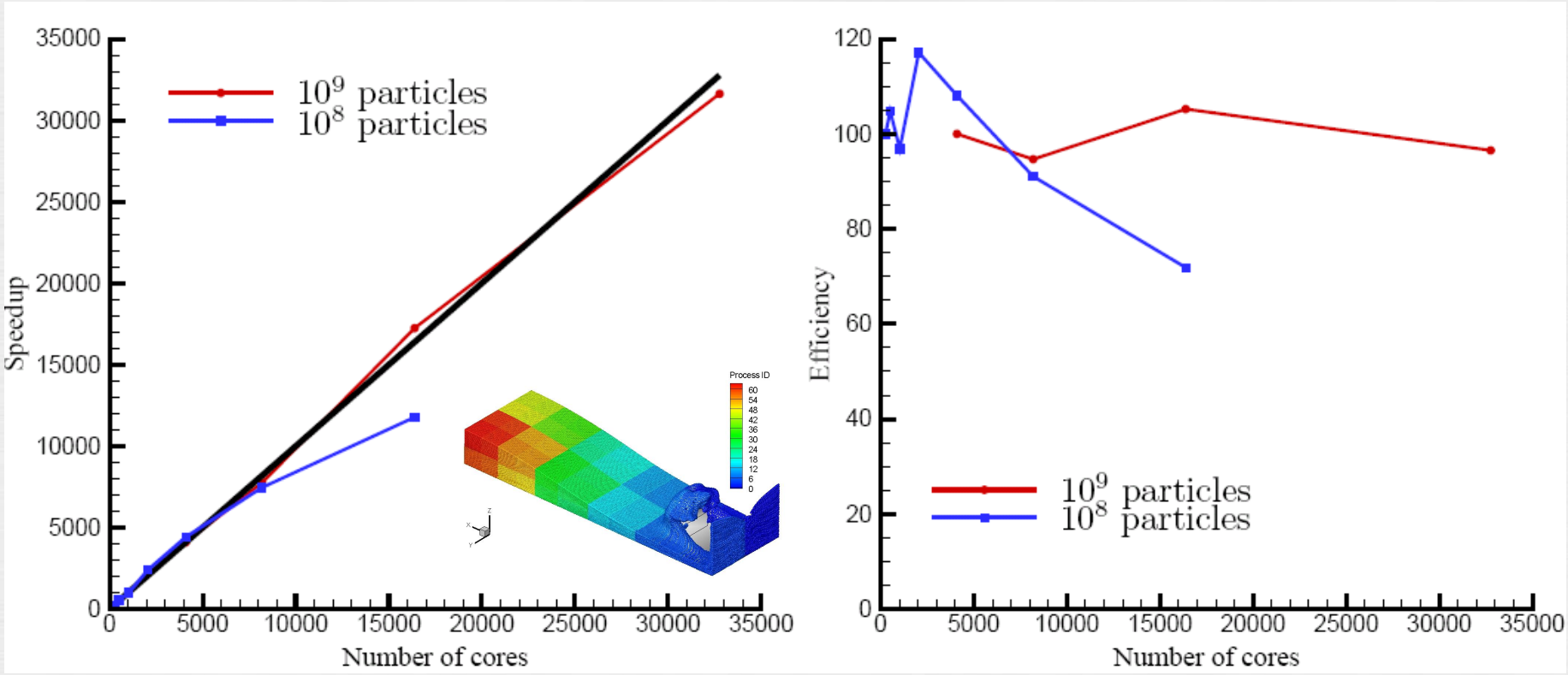


Masquage des temps de communications au cours du calcul

- Diagramme simplifié de fonctionnement parallèle du code SPH-Flow



Performances parallèles pour un problème à taille fixée (« Strong scalability »)

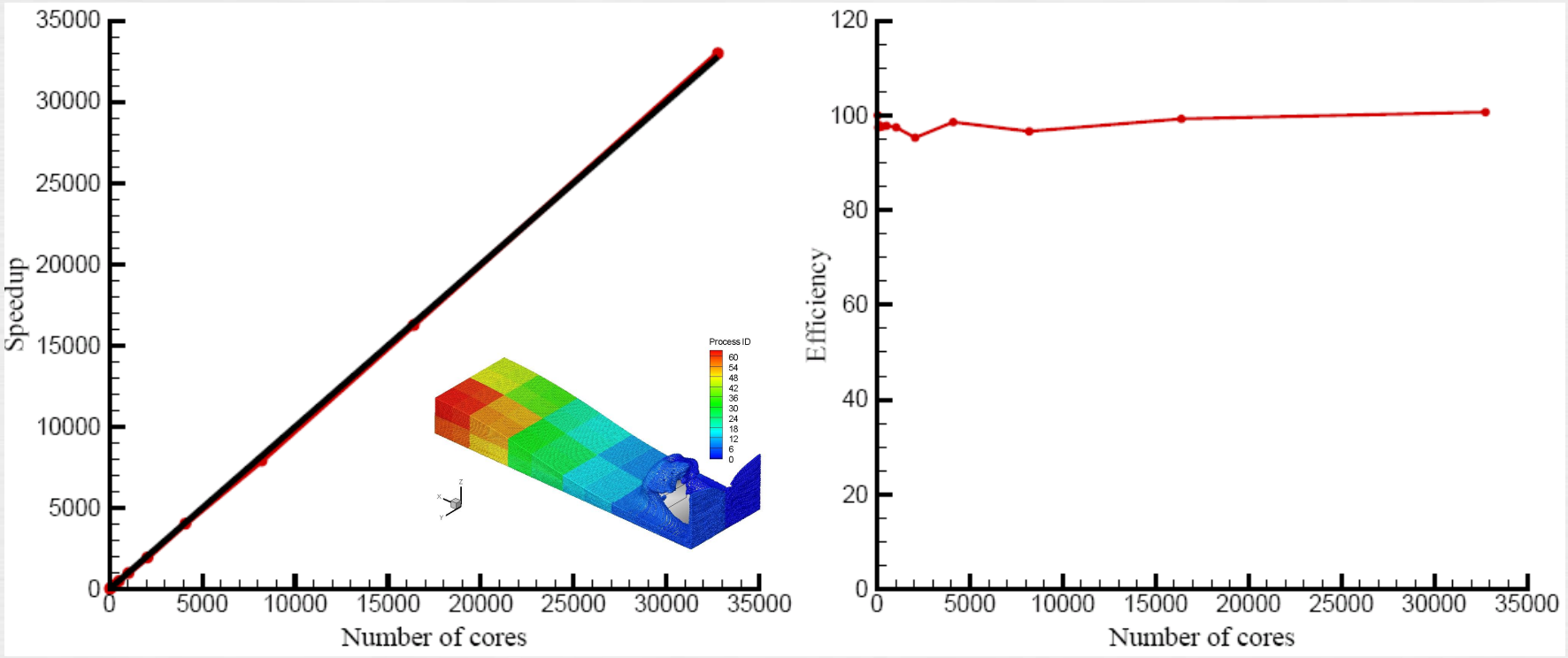


Performances obtenues pour 100 millions et 1 milliard de particules (jusqu'à 32 768 cœurs)

- Bonnes performances dans l'ensemble
- Quelques imperfections dues à certaines parties demeurant séquentielles
- Absence de rupture de pente du speedup (code scalable)

Performances parallèles à charge fixée par cœur (« Weak scalability »)

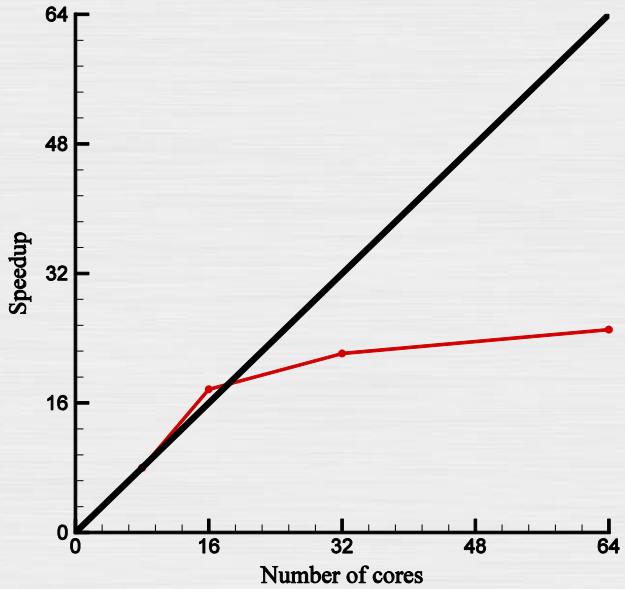
Charge: 10^5 particules/cœur

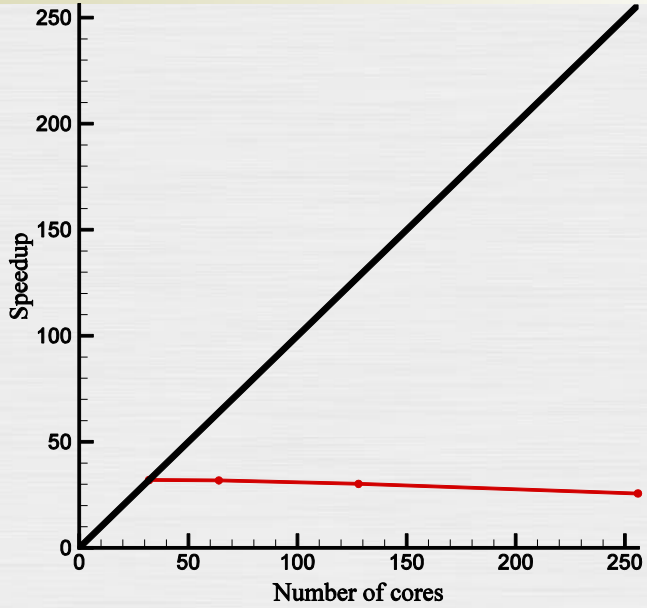


Performances obtenues jusqu'à 3,5 milliards de particules et jusqu'à 32 768 cœurs

- Absence de bottleneck sur les communications

IO par utilisation de HDF5 parallèle (tests effectués sur système Lustre)





Number of cores	8	16	32	64
Sequential ASCII	428.6 s	447.4 s	426.9 s	432.5 s
Parallel HDF5	36.5 s	16.5 s	13.2 s	11.6 s
Speedup (sequential ASCII/parallel HDF5)	11.7	27.1	32.3	37.3

Ecritures pour 66 millions de particules

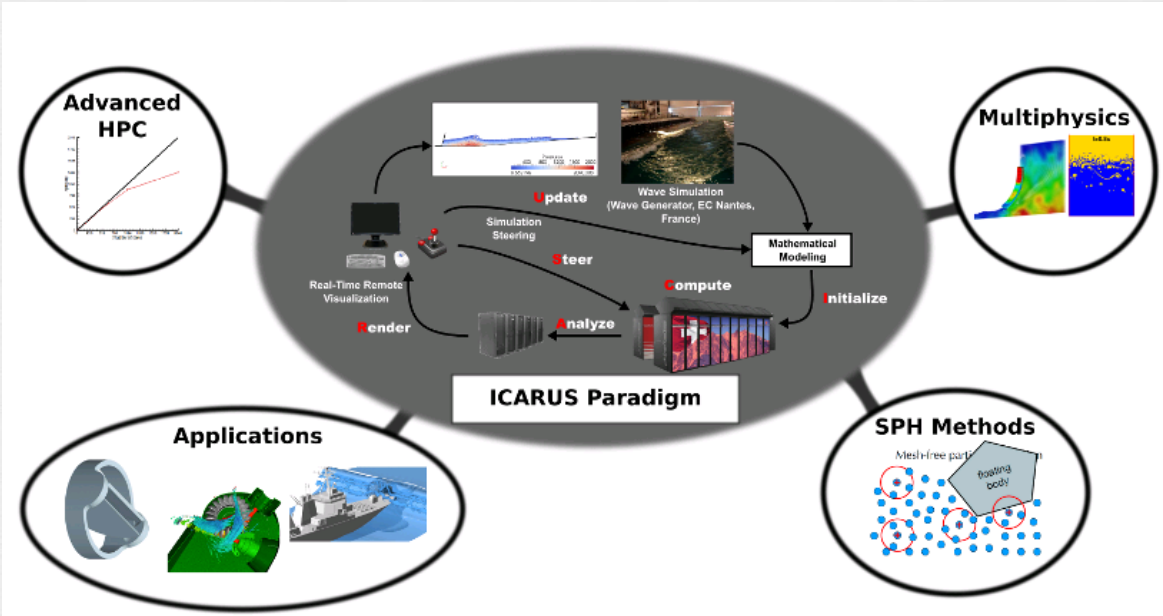
Number of cores	32	64	128	256
Sequential ASCII	5060 s	5102 s	5070 s	5090 s
Parallel HDF5	112 s	113 s	119 s	140 s
Speedup (sequential ASCII/parallel HDF5)	45	45	42.6	36.3

Ecritures pour 770 millions de particules

- Gains très importants par rapport à l'ASCII
- Performances limitées pour des grands nombre de particules, et p>64 cœurs

Inter-opérabilité: Robot ICARUS
Projet de recherche européen NextMuse http://nextmuse.cscs.ch

- ICARUS (Initialize-Compute-Analyze-Render-Update-Steer)
- Simulation interactive en temps réel sur des simulations massives (≈10^8 particules et ≈2000 cœurs)
- Utilisation de HDF5-DSM: communication de données entre un réseau de processeurs de calcul et un réseau de processeur de rendu graphique par lecture/écriture de fichiers virtuels HDF5
- Partenariat avec le CSCS (Swiss Supercomputing Center)



Perspectives d'amélioration HPC (initiative HPC-PME)

- Architectures à mémoire partagé?
- Accélérateurs: GPGPU ou Intel Xeon PHI?

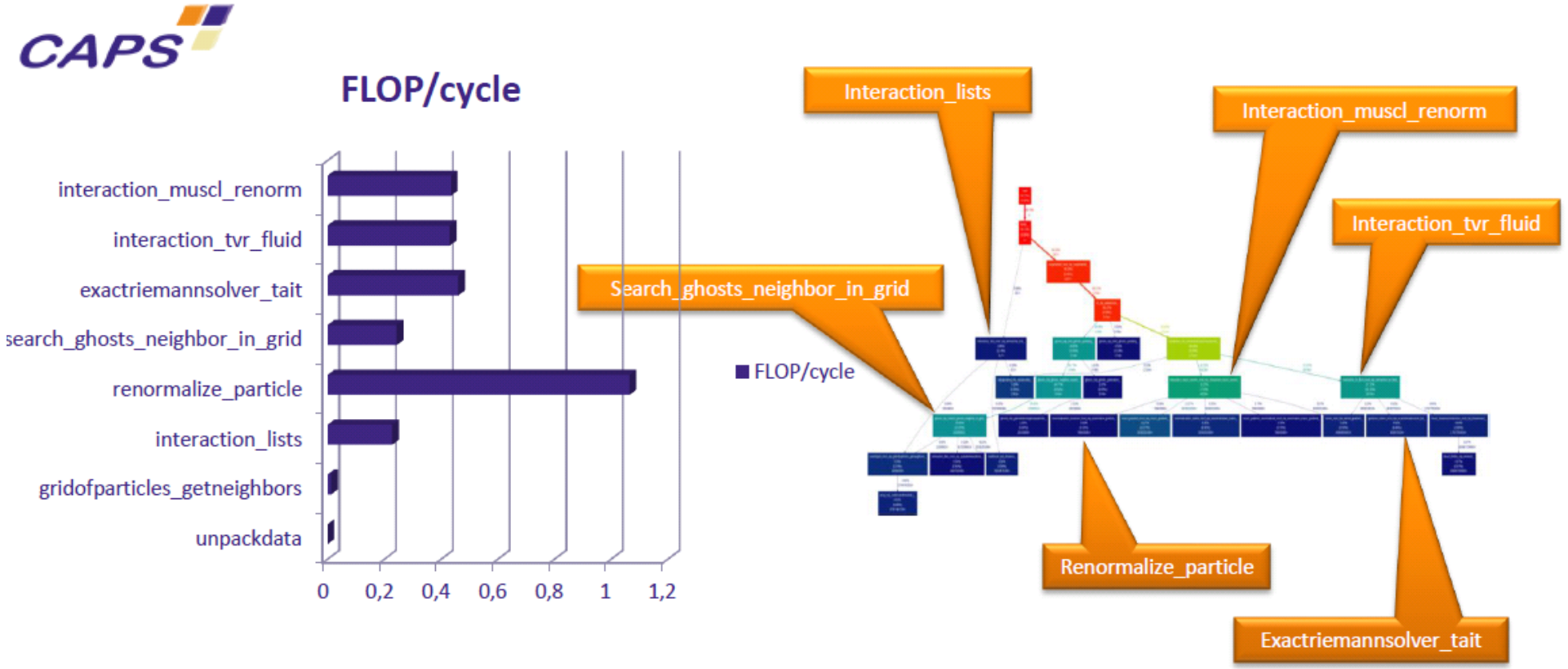
- portage de SPH-Flow sur les architectures GPGPU
- gain: 20 à 30
- 2 approches
  - Écriture native (CUDA) -> réécriture complète de notre code Fortran
  - Directives à la OpenMP: OpenACC
- solution technique retenue: par directives OpenACC
  - peu intrusives
  - #pragma acc …

- But de notre projet HPC-PME:
  - Développer le code pour ces nouvelles architectures
  - Réduire fortement les temps de nos calculs

Profiling

- hotspot = 60% => speedup 2 à 3 (Amdahl)
- Pourquoi pas 20 ou 30 comme dans la littérature ?
  - GPU: ~1TFlops
  - 1coeur: 2.5GFlops
  - 32coeurs: 80GFlops
  - 32coeurs with SSE (4Flops per cc): 320GFlops
  - 32coeurs with AVX (8): 640GFlops
- ratio: seulement 3 entre GPGPU et version optimisée parallèle sur CPU
- Questions:
  - Intérêt du GPGPU? pour un industriel x2 pas négligeable
  - OpenMP ou OpenACC?

Efficacité vectorielle



- Les CPU supportent additions et multiplications vectorielles
  - SSE, AVX
  - 4 à 8 octets simultanés
  - voire même une addition et une mutliplication en même temps

- Notre code nécessite un meilleure vectorisation (x4 à x8 possible)

Algorithmie adaptée

Points d'attention

- SPH est sans maillage MAIS repose sur une grille.
- simulations
  - à discrétisation constante
  - à discrétisation variable
- MPI avec une « couche » de recouvrement

Solutions retenues

- Grille régulière et hiérarchique sur GPGPU.
- Algorithme de tri (radixsort)
- Algorithme de recherche (binary search)

Points durs de l'hybridation

- Arrays of Structure converti en Structure of Arrays
  - AoS: particles(component, id)
  - SoA: particles%component(id)
- Voisinage par morceaux: meilleure utilisation du cache
  - Casser les indirections
  - Contiguïté en mémoire
- Grille hiérarchique linéaire
  - Élement central

- 90% du code impacté
- Restructuration avant portage pour une meilleure vectorisation
- Hybridation MPI/OpenMP pour valider le refonte
- Mise en place des directives OpenACC.

Conclusion

- SPH: méthode particulaire Lagrangienne permettant des simulations multiphysiques très complexes
- Parallélisation sur mémoire distribuée MPI efficace et scalable (scalabilité testée jusqu'à 3,5 milliard de particules et 32 768 cœurs)
  - Passage Tier-2 vers Tier-0 en 1an
  - 2 projets PRACE (5millions d'h sur Hermit et 8 millions d'h sur Curie)
- Permet des simulations massives pour des cas complexes
- Des parties séquentielles demeurent et doivent être parallélisée
- Les temps de restitution demeurent élevés sur CPUs, un nombre minimum de 10 000 particules par cœurs en 3D est nécessaire pour être scalable
- Le portage GPU peut permettre de réduire les temps de restitution.