

Application du calcul intensif sur GPU aux simulations mésoscopiques de suspensions colloïdales

M. Cerbelaud^a, A. Videcoq^a, J. Gerhards^a,
B. Crespin^b

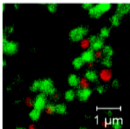
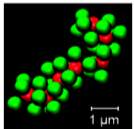
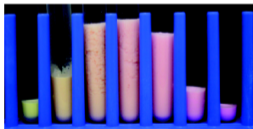
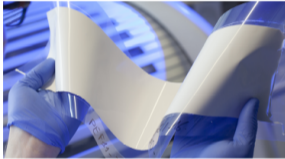
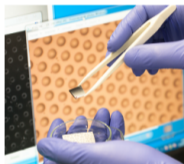
a) University of Limoges, CNRS, IRCER, UMR 7315, France

b) University of Limoges, CNRS, XLIM, UMR 7252, France

Projet collaboratif : Xlim / IRCER

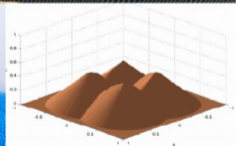
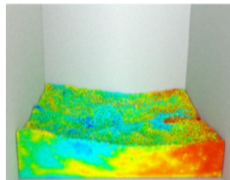
irCer
institut de recherche
sur les céramiques

Axe 1

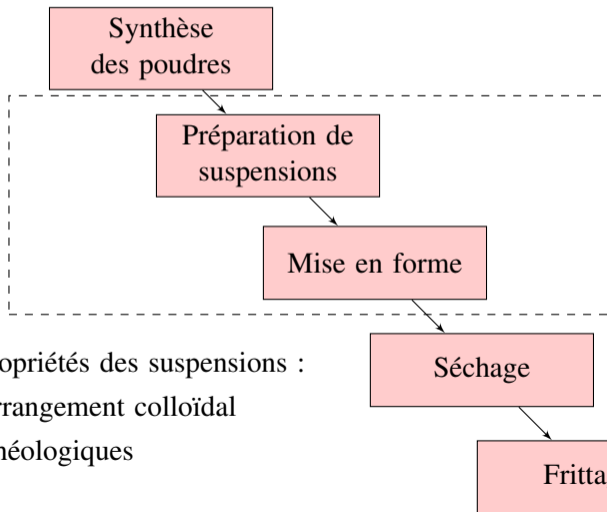


xlim INSTITUT
DE RECHERCHE

ASALI



Principales étapes d'un procédé céramique par voie colloïdale



Contrôle des propriétés des suspensions :

- ▶ Structure, arrangement colloïdal
- ▶ Propriétés rhéologiques

Simulations de suspensions colloïdales utilisées à l'IRCER

Dynamique Brownienne (DB)

Fluide : milieu continu

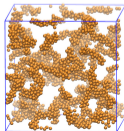
Dynamique des colloïdes :

$$m \frac{dv}{dt} = -\zeta v + \sum F(r) + \Gamma(t)$$

Force de
friction

Interactions
entre
particules

Force
aléatoire



Pas d'interaction
hydrodynamique

'Stochastic rotation dynamics- molecular dynamics' (SRD-MD)

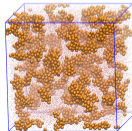
Fluide : particules avec dynamique simple

$$\text{Ecoulement : } x(t + dt) = x(t) + v(t)dt$$

$$\text{Collision : } v(t + dt) = v_{cm}(t) + R(v(t) - v_{cm})$$

Colloïdes et couplage : MD

$$m \frac{dv}{dt} = \sum F(r)$$

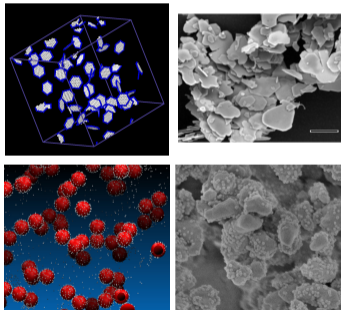


Bonne description
des interactions
hydrodynamiques

Développement des simulations sur CPU

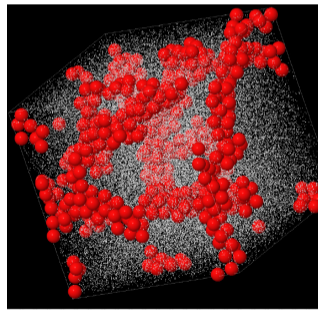
Développement des codes en interne et exécution en séquentiel sur 1 CPU

DB



Quelques milliers de colloïdes

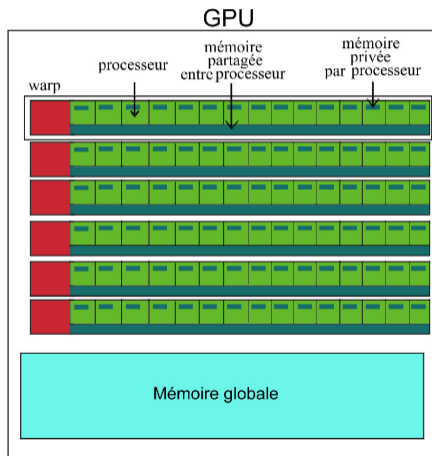
SRD-MD



Quelques centaines de colloïdes

→ Augmentation de la taille des systèmes

Parallélisation des codes sur GPU : collaboration avec Xlim (Labex Σ Lim)



-Grand nombre d'unité de calculs permettant d'effectuer des opérations logiques et arithmétiques

-Exécution simultanée de la même fonction par plusieurs cœurs sur différentes données.

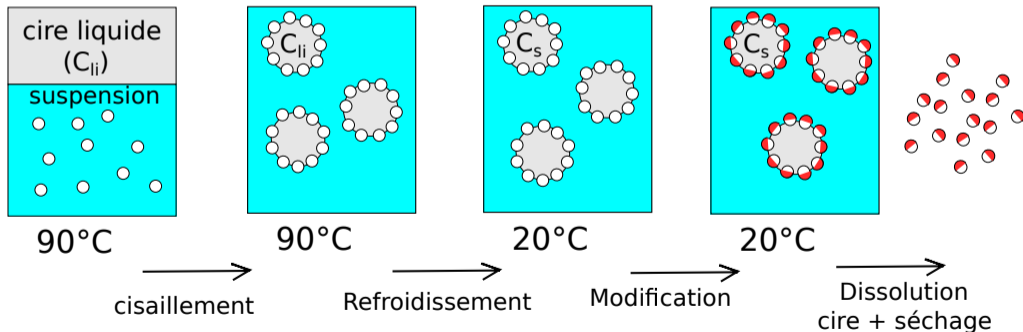
→ Intéressant pour la dynamique moléculaire

→ Premier codes développés avec OpenCL

Application à l'étude de particules Janus

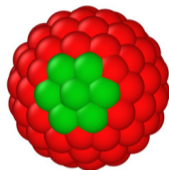
But : Utiliser l'anisotropie de surface pour créer des arrangements colloïdaux originaux

Synthèse de particules :

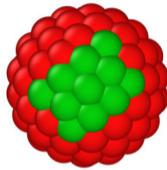


La grandeur du patch est définie par l'enfoncement des particules dans la cire

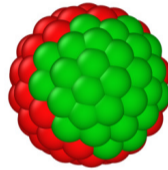
Modélisation des particules 'zwitterioniques'



S=92%



S=88%



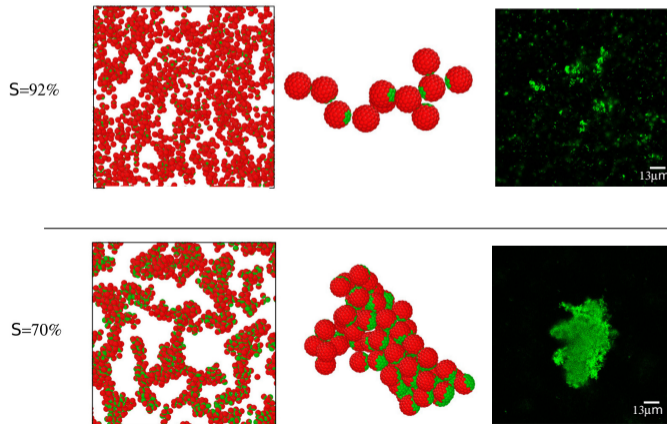
S=70%

- 1 particule : sphère creuse indéformable décorée de 92 sphères élémentaires.
- Sur 1 CPU : avec un pas de temps de 10^{-7} s et 50 particules
 - plus d'un mois pour simuler 10 s
 - **Adaptation des codes GPU pour cette étude.**

Résultats des simulations de particules amphotères

Carte GTX 1080Ti (3584 cœurs, 11Gb, 2018) avec un pas de temps de 10^{-7} s et 1500 particules

→ pour 10 s : 6 jours

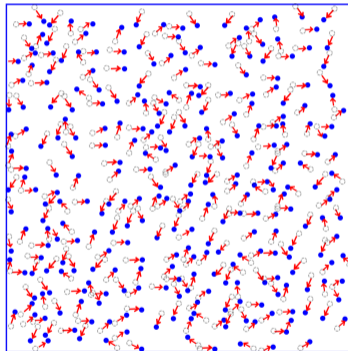


Stochastic Rotation Dynamics (SRD) : description du fluide

Dynamique du fluide :

Écoulement :

$$x(t + dt) = x(t) + v(t)dt$$



Stochastic Rotation Dynamics (SRD) : description du fluide

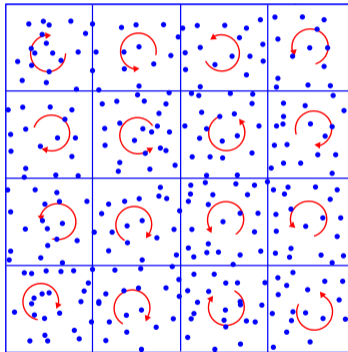
Dynamique du fluide :

Écoulement :

$$x(t + dt) = x(t) + v(t)dt$$

Collision :

$$v(t + dt) = v_{cm}(t) + R(v(t) - v_{cm})$$

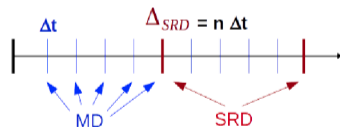
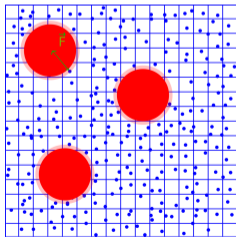


Stochastic Rotation Dynamics-Molecular dynamics (SRD-MD) : suspensions colloïdales

Fluide : SRD

Colloïdes : MD

Couplage : Interactions entre les
particules de fluide et les colloïdes

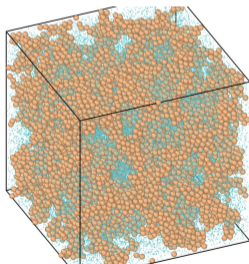


1 itération = n étapes de MD + 1 étape de SRD

Comparaison de cartes graphiques

- GTX 1080Ti avec 3 584 cœurs et 11GB de mémoire (2018 - CALI2)
- P100 avec 3584 cœurs et 16GB de mémoire (2016 - CURTA)
- RTX2080 avec 2 944 cœurs et 8GB de mémoire (2020)
- A40 avec 10 752 cœurs et 48GB de mémoire (2021 - YAGA)

Temps moyen par itération (ms) : moyenne sur 55 000 itérations



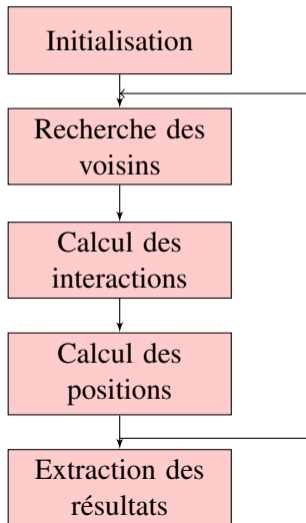
4000 colloïdes à 10v%

(attraction type Lennard Jones 36-18)

Nb colloïdes	Nb particules de fluide	GTX 1080	RTX 2080	P100 Curta	A40 Yaga
500	788 981	145	131	119	540
2 000	3 120 942	213	193	178	636
4 000	6 311 854	298	273	244	702
8 000	12 623 708	532	462	388	800
20 000	31 507 631	xxxx	xxxx	xxxx	1342
40 000	63 128 808	xxxx	xxxx	xxxx	2328

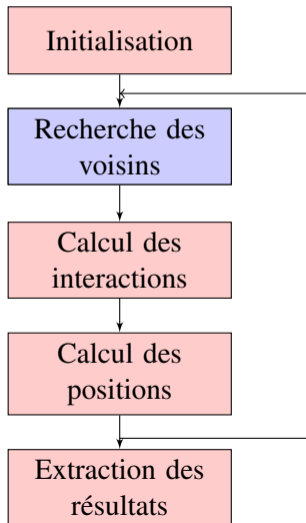
→ Paramètre qui peut être limitant : la mémoire

Étapes principales des programmes de MD :



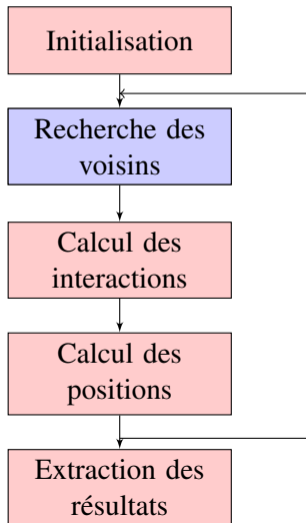
Parallélisation des calculs sur GPU
Langage OpenCL

Étapes principales des programmes de MD :



Parallélisation des calculs sur GPU
Langage OpenCL

Étapes principales des programmes de MD :



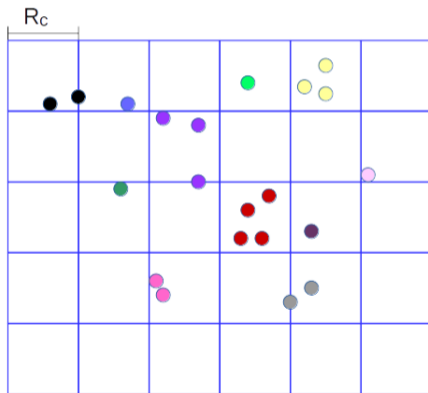
Parallélisation des calculs sur GPU Langage OpenCL

Pour les interactions entre colloïdes :
utilisation de potentiels à courte
portée avec un rayon de coupure R_c

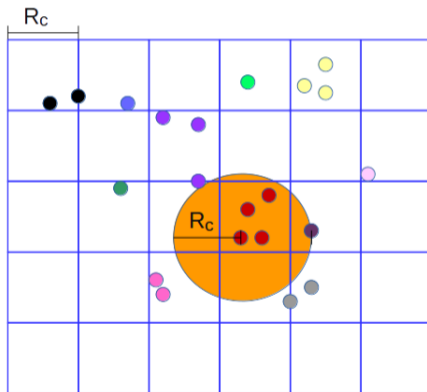
Recherche de Voisins : Grille régulière



Recherche de Voisins : Grille régulière



Recherche de Voisins : Grille régulière

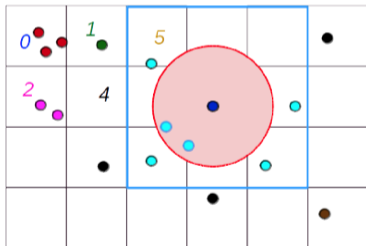


taille cellule $\geq R_c$

Recherche de Voisins : Grille régulière

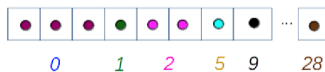
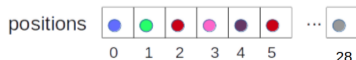
Besoin de trier les tableaux : 80% du temps avec une grille régulière

Grille régulière



Z-index

X:	0	1	2	3
	000	001	010	011
Y: 0	000000	000001	000100	000101
0	000	0	1	5
1	000010	000011	000110	000111
1	001	2	4	
2	001000	001001	001100	001101
2	010			
3	001010	001011	001110	001111
3	011			



Recherche de Voisins : Liste de Verlet

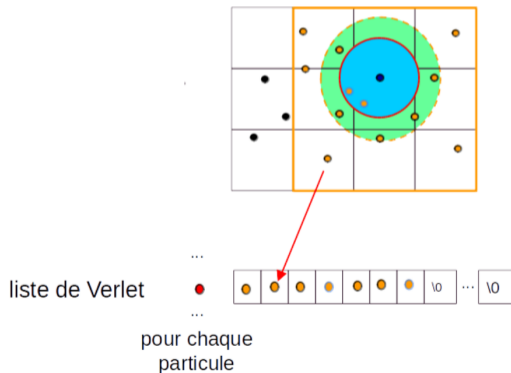


- Nécessité de calculer toutes les distances pour faire les listes
- Le test de reconstruction des listes doit être simple
(Pas de somme des deux plus grands déplacements)

→ Test moins performant : $d_{max} > R_s/2$

Recherche de Voisins : Mixte entre liste de Verlet et grille régulière

Utilisation de la grille régulière pour construire les listes de Verlet

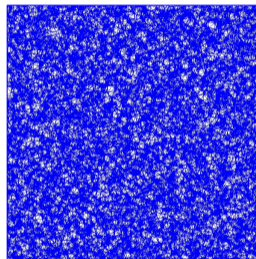
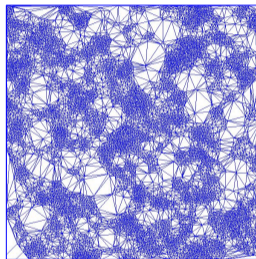
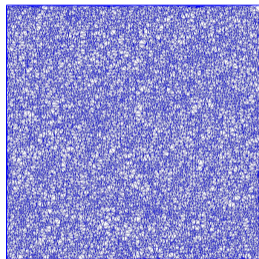


Conclusion

- Parallélisation GPU des codes de **dynamique Brownienne** :
 - Recherche de voisins : grille + liste de Verlet
 - milliers de particules → **millions de particules**
 - Adaptation à des codes complexes avec des **particules discrétisées**
 - Stratégie adaptable à la dynamique moléculaire
- Parallélisation GPU des codes de **SRD-MD**
 - Adaptation du schéma de décomposition
 - centaines de particules → **milliers de particules**
 - Limite : mémoire

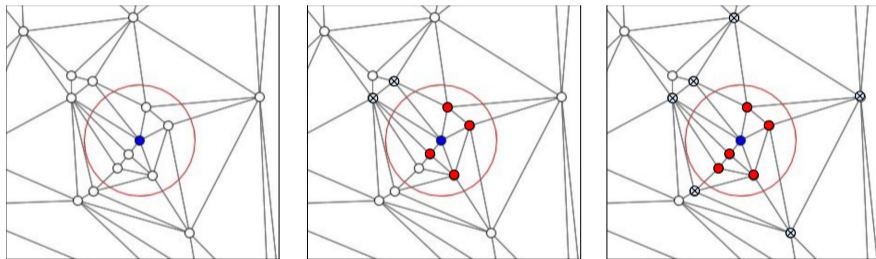
Remarque : Architecture des cartes graphiques et des versions de CUDA/OPENCL en permanente évolution → modification des codes en conséquence

Optimisation de la Recherche de Voisins : Projet ANR SOMA-DNS



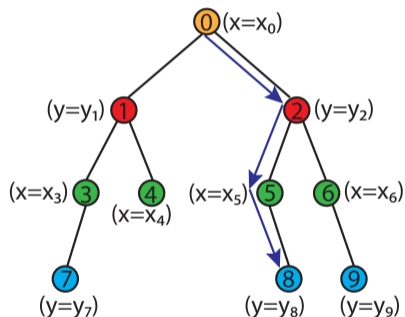
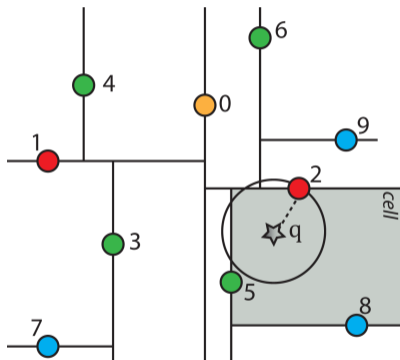
- SOft MATter dynamics with Delaunay-based Neighbours Search
- Mise à jour de la triangulation après déplacement des sommets

Optimisation de la Recherche de Voisins : Projet ANR SOMA-DNS



- Recherche de voisinage dans le graphe (Breadth-First Search)
- "Fixed-radius Near Neighbors Searching for 2D Simulations on the GPU using Delaunay Triangulations", H. Porro, B. Crespin, N. Hitschfeld-Kahler, C. Navarro, Eurographics 2022 - Posters

Optimisation de la Recherche de Voisins : modèles hiérarchiques



Exemple de recherche dans un KD-tree (Li et al 2014)

Merci pour votre attention !

Remerciements :



Xavier Montagutelli
Equipe CALI et YAGA
Equipe MCIA
Florent Di-Meo
Khaoula Lebdioua
Anne Aimable
Claire Carrion