

Introduction d'un modèle de spray dans le code MaHyCo

24 mars 2025 – Journée Arcane

Lucas Prouvost

CEA-DIF

Travaux réalisés dans le cadre du LRC MESO pendant l'été 2024

Plan

- I) MaHyCo
- II) Interaction particules/gaz
- III) MaHyCo, particules et Arcane
- IV) Résultats



MaHyCo

MaHyCo: Maquette Hydrodynamique Collaborative

Maquette *open-source* pour tests et validations de méthodes numériques (https://github.com/cea-hpc/MaHyCo, hors particules) – initiée en 2020 – licence Apache 2.0

Basée sur la plateforme Arcane

Résolution des équations de la mécanique des fluides compressibles multi-matériaux

Missions du code :

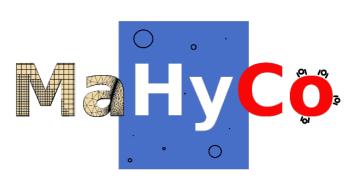
Utilisé pour les stages, post-doc et télétravail (et thèses prochainement ?)

Prototype d'un code hydrodynamique sur GPU

Pré-industrialisation (non-régression)

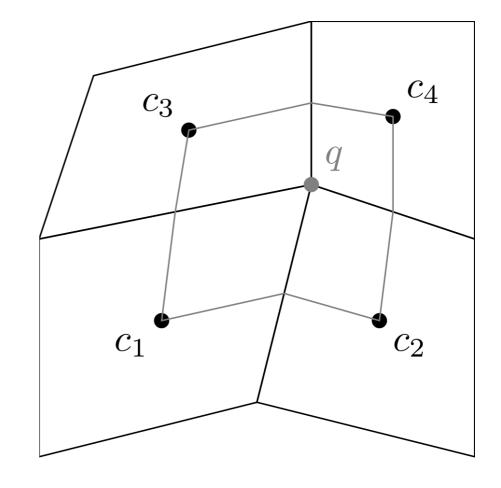
Objectif du stage :

Ajout d'un modèle de spray



MaHyCo

- Fluides
- Multi-matériaux (iso-déformation)
- Schéma « VNR » [1,2,3] :
 - « Lagrange + projection »
 - Chocs : viscosité artificielle
- Volume finis
- Staggered in time and space :
 - Vitesse aux nœuds décalée en temps $oldsymbol{u}_q^{n+1/2}$
 - Autres variables *cell-centered* ρ^n , m^n , e^n , ...



Cellules primales $(c_i)_{i \in [1,4]}$ et duale q

MaHyCo: équations pour les fluides

Formulation Lagrangienne

Milieu i

Masse et masse nodale m_i , m_q

Vitesse nodale u_q

Force de pression nodale F_q

Énergie interne e_i

Pression p_i

Viscosité artificielle q_i

Volume spécifique $v = \frac{1}{\rho}$

Position des nœuds x_q

$$\frac{Dm_i}{Dt} = 0$$

$$\frac{D\boldsymbol{u}_q}{Dt} = \frac{1}{m_q} \boldsymbol{F}_q$$

$$\frac{De_i}{Dt} = -(p_i + q_i) \frac{Dv_i}{Dt}$$

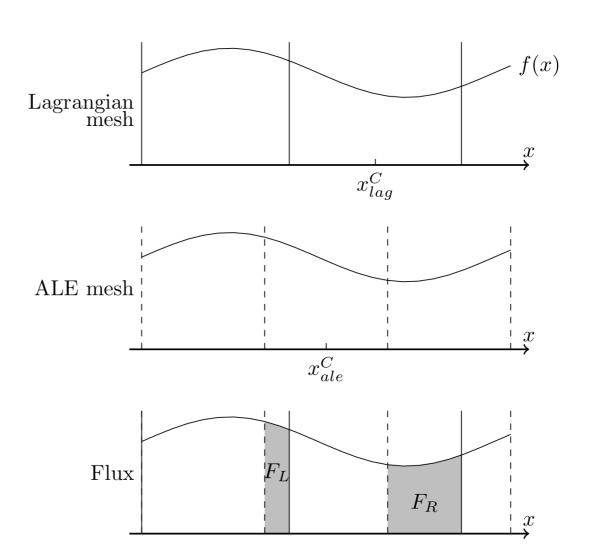
$$\frac{D\boldsymbol{x}_q}{Dt} = \boldsymbol{u}_q$$

$$p_i^{n+1} = EOS(\rho_i^{n+1}, e_i^{n+1})$$

MaHyCo: étape de projection/remap

- Projection à direction alternée (ADI) :
 - 2D/3D = suite de projection 1D
 - Strang splitting (ordre 2)
- Multi-matériaux : pas de reconstruction d'interface pour le moment (projet futur prévu)
- Calcul des flux F_L, F_R avec possibilité de
 limiteurs de gradient (ordre 1, 2 ou 3 en espace)

$$f\left(x_{ale}^{C}\right)V_{ale}^{C} = f\left(x_{laq}^{C}\right)V_{laq}^{C} + F_{L} - F_{R}$$



Interaction particules – gaz / sprays

Problèmes industriels

- Moteurs diesel
- Propulseurs à poudre
- Pollution atmosphérique
- Biologie

Modèles particules-gaz

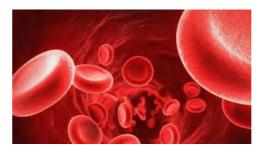
O'Rourke, Cloutman, Dukowicz

Code de calcul

KIVA (Los Alamos) 1979 [4]









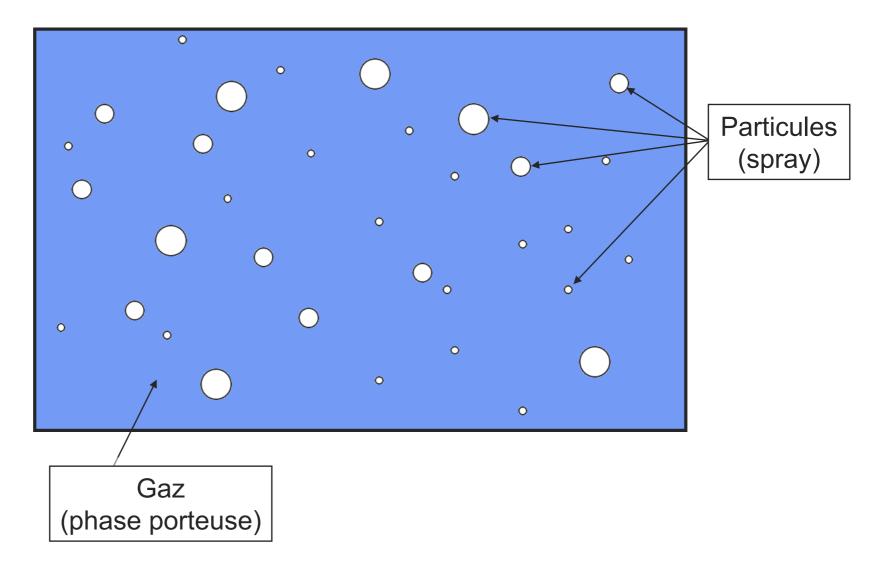
Problème type

3 approches possibles:

DNS

Multiphasique (Euler/Euler)

 Méthode particulaire (Euler/Lagrange)

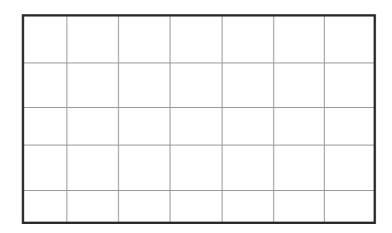


Méthode particulaire (Euler/Lagrange)

- 1 phase résolue
- Particules individuelles à suivre
- Équations monophasique
 - + termes d'interaction fluide/particules

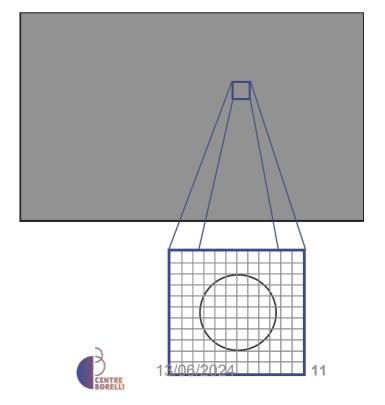
Approche multiphasique (Euler/Euler)

- 2 phases résolues
- Comportement moyen des phases
- Mélange homogène



Approche DNS

- 2 phases résolues
- Petites échelles

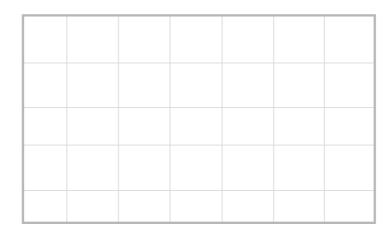


Méthode particulaire (Euler/Lagrange)

- 1 phase résolue
- Particules individuelles à suivre
- Équations monophasique
 - + termes d'interaction fluide/particules

Approche multiphasique (Euler/Euler)

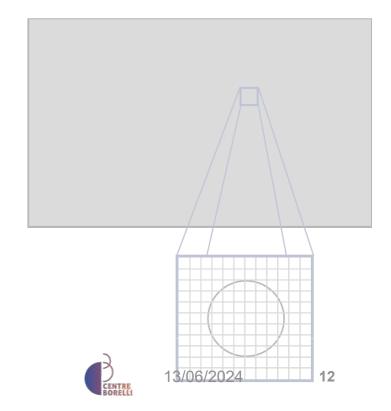
- 2 phases résolues
- Comportement moyen des phases
- Mélange homogène



- - → particules avec des caractéristiques différentes (taille, vitesse, ...)
 - → moins diffusif

Approche DNS

- 2 phases résolues
- Petites échelles



Objectif : décrire **finement** l'**interaction** de particules et d'un gaz



Modèle à phase dispersée :

Mélange de 2 types de représentation (approche « Euler-Lagrange »)

- Continue pour le fluide porteur
- Discrète pour les particules

Modèle couplé :

- Systèmes d'équations couplés
- Prise en compte de la rétroaction des particules sur le gaz

Représentation discrète des particules :

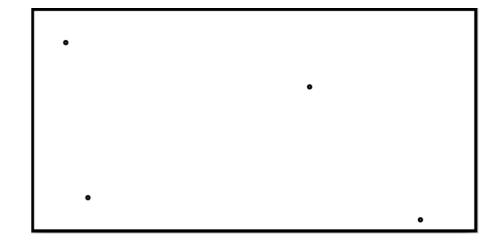
Une particule (numérique) = Une classe contenant N_p particules physiques ayant des *propriétés similaires* dans l'espace des phases

$$x_p \in [x, x + dx], \ u_p \in [u, u + du], \ r_p \in [r, r + dr], \dots$$

Classification par niveau d'interaction entre le fluide et les particules [6,7]

Sprays très fins:

- Fraction volumique des gouttes/particules négligeable
- Masse des gouttes non-négligeable



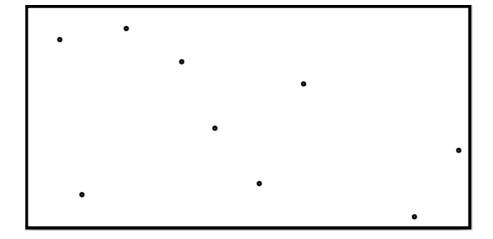
Classification par niveau d'interaction entre le fluide et les particules [6,7]

Sprays très fins :

- Fraction volumique des gouttes/particules négligeable
- Masse des gouttes non-négligeable

Sprays fins:

Interaction fluide-particules via un terme de traînée



Classification par niveau d'interaction entre le fluide et les particules [6,7]

Sprays très fins:

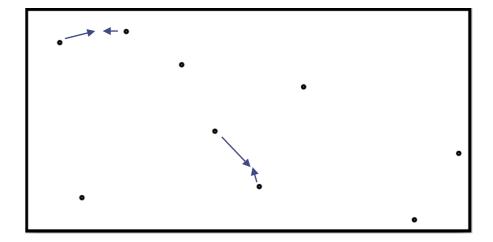
- Fraction volumique des gouttes/particules négligeable
- Masse des gouttes non-négligeable

Sprays fins:

Interaction fluide-particules via un terme de traînée

Sprays moyennement épais :

Interaction entre particules (collision, coalescence, ...)



Classification par niveau d'interaction entre le fluide et les particules [6,7]

Sprays très fins:

- Fraction volumique des gouttes/particules négligeable
- Masse des gouttes non-négligeable

Sprays fins:

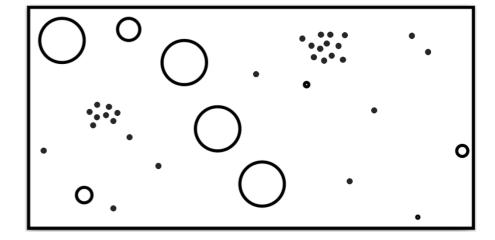
Interaction fluide-particules via un terme de traînée

Sprays moyennement épais :

Interaction entre particules (collision, coalescence, ...)

Sprays épais :

- Fraction volumique des particules non-négligeable
- Poussée d'Archimède due aux gradients de pression non nécessairement négligeable devant la force de traînée



Classification par niveau d'interaction entre le fluide et les particules [6,7]

Sprays très fins:

- Fraction volumique des gouttes/particules négligeable
- Masse des gouttes non-négligeable



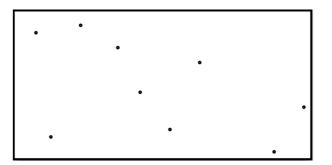
Interaction fluide-particules via un terme de traînée

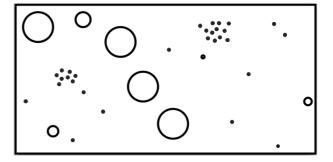
Sprays moyennement épais :

Interaction entre particules (collision, coalescence, ...)

Sprays épais :

- Fraction volumique des particules non-négligeable
- Poussée d'Archimède due aux gradients de pression non nécessairement négligeable devant la force de traînée





Sprays très fins

Équations du fluide

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \rho \mathbf{g}$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = -p \nabla \cdot (\mathbf{u}) + \nabla \cdot (K\nabla T)$$

$$EOS$$

Équations des particules

$$\frac{D\boldsymbol{x}_p}{Dt} = \boldsymbol{u}_p$$

$$m_p \frac{D\boldsymbol{u}_p}{Dt} = m_p \, \boldsymbol{g} + D_p (\boldsymbol{u} - \boldsymbol{u}_p)$$

+ autres paramètres de l'espace des phases (température, rayon, etc)

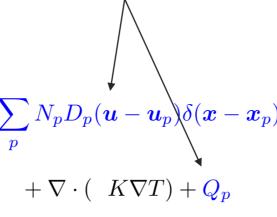
Traînée :
$$D_p = rac{3\,
ho\,C_D\,m_p}{8\,
ho_p\,r_p}|oldsymbol{u}-oldsymbol{u}_p|$$

Sprays fins

Equations du fluide

 $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$ $\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \rho \mathbf{g} - \sum_{p} N_{p} D_{p} (\mathbf{u} - \mathbf{u}_{p}) \delta(\mathbf{x} - \mathbf{x}_{p})$ $\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = -p \nabla \cdot (\mathbf{u})$ $+ \nabla \cdot (K \nabla T) + Q_{p}$ $\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = -p \nabla \cdot (\mathbf{u})$ $+ \nabla \cdot (K \nabla T) + Q_{p}$

Interaction particules – gaz



Equations des particules

$$\frac{D\boldsymbol{x}_p}{Dt} = \boldsymbol{u}_p$$

$$m_p \frac{D\boldsymbol{u}_p}{Dt} = m_p \, \boldsymbol{g} + D_p (\boldsymbol{u} - \boldsymbol{u}_p)$$

+ autres paramètres de l'espace des phases (température, rayon, etc)

Q_p : échange de chaleur fluide-particules

Traînée :
$$D_p = rac{3\,
ho\,C_D\,m_p}{8\,
ho_p\,r_p}|oldsymbol{u}-oldsymbol{u}_p|$$

EOS

22

MaHyCo, particules et Arcane



- Suivre des objets Lagrangiens
- Dans une simulation eulérienne

Particle particle = *ipart;

- Suivre des objets Lagrangiens
- Dans une simulation eulérienne
- Objets indépendants du maillage
- Mais localisables dans le maillage (rétroaction)

Particle particle = *ipart;

```
m_particles_family->setParticleCell(particle, cell);

ENUMERATE_PARTICLE (ipart, activeParticlesGroup) {
   Particle particle = *ipart;
   Cell cell = particule.cell();
```



- Suivre des objets Lagrangiens
- Dans une simulation eulérienne
- Objets indépendants du maillage
- Mais localisables dans le maillage (rétroaction)
- Variables associées à l'objet (position, vitesse, rayon...)
- Boucles sur ces objets
- Familles et Groupes (actives, hors-domaine, ...)
 - → Items

```
Particle particle = *ipart;
```

```
m_particles_family->setParticleCell(particle, cell);

ENUMERATE_PARTICLE (ipart, activeParticlesGroup) {
   Particle particle = *ipart;
   Cell cell = particule.cell();

   Real3 particule_coord = m_particle_coord[ipart];

   IItemFamily* m_item_family;
        IParticleFamily* m_particles_family;
        ParticleGroup m_activeParticlesGroup;
}
```

```
m_item_family = mesh()->createItemFamily (eItemKind::IK_Particle, "AllParticles");
m_activeParticlesGroup = m_item_family->createGroup("activeItem");
m_particles_family = m_item_family->toParticleFamily();
```

- Suivre des objets Lagrangiens
- Dans une simulation eulérienne
- Objets indépendants du maillage
- Mais localisables dans le maillage (rétroaction)
- Variables associées à l'objet (position, vitesse, rayon...)
- Boucles sur ces objets
- Familles et Groupes (actives, hors-domaine, ...)
- Parallélisme (changement de sous-domaines)
- Protection/Reprise (gérées par Arcane)
- Entrées/Sorties (fichier utilisateur, ...)

Particle particle = *ipart;

```
m_particles_family->setParticleCell(particle, cell);

ENUMERATE_PARTICLE (ipart, activeParticlesGroup) {
   Particle particle = *ipart;
   Cell cell = particule.cell();
```

Real3 particule_coord = m_particle_coord[ipart]

```
IItemFamily* m_item_family;
IParticleFamily* m_particles_family;
ParticleGroup m_activeParticlesGroup;
```

```
m_item_family = mesh()->createItemFamily (eItemKind::IK_Particle, "AllParticles");
m_activeParticlesGroup = m_item_family->createGroup("activeItem");
m_particles_family = m_item_family->toParticleFamily();
```

m_particles_family->endUpdate();

IParticleExchanger* _getParticleExchanger(){return options()->getParticleExchanger();}





Déplacement d'une particule :

$$m{x}_p^{n+1} - m{x}_p^n = dt^{n+1/2} \, m{u}_p^{n+1/2}$$

```
void SprayFinService::updateParticlePosition() {
   const Real dt (m_global_deltat()); // dt^{n+1/2}
   ENUMERATE_PARTICLE (part_i, activeParticlesGroup) {
      m_particle_coord[part_i] += m_particle_velocity[part_i] * dt;
   }
}
```

Rétroaction fluide/particules :

```
Node SprayFinService::findNodeOfParticle(Particle particule){
   Cell cell = particule.cell();

// la particule appartient au noeud le plus proche
   Real distance_min=le30;
   Node node_proche;
   for ( NodeEnumerator inode ( cell.nodes() ); inode.hasNext(); ++inode ) {
      Real distance = (m_node_coord[inode] - m_particle_coord[particule]).normL2();
      if (distance < distance_min){
            distance_min = distance;
            node_proche=*inode;
      }
   }
   return node_proche;
}</pre>
```

Rétroaction fluide/particules :

```
void SprayFinService::updateParticleVelocity() {
   const Real dt ( 0.5 * ( m_global_old_deltat() + m_global_deltat() ) ); // dt^n

ENUMERATE_PARTICLE (ipart, activeParticlesGroup) {
   Real3 up_chapo = m_particle_velocity[ipart] + dt*options()->getGravity();
   Particle particle = *ipart;
   Node node_proche = findNodeOfParticle(particle); // cherche à quel noeud appartient la particule
   Real3 Dp = computeDp(particle, node_proche);
   Real3 one={1., 1., 1.};

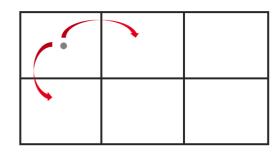
   m_particle_velocity[ipart] = ( up_chapo + dt*Dp*m_velocity[node_proche] )/( one + dt*Dp );
}
```

```
Node SprayFinService::findNodeOfParticle(Particle particule){
    Cell cell = particule.cell();

    // la particule appartient au noeud le plus proche
    Real distance_min=le30;
    Node node_proche;
    for ( NodeEnumerator inode ( cell.nodes() ): inode.hasNext(): ++inode ) {
        Real distance = (m_node_coord[inode] - m_particle_coord[particule]).normL2();
        if (distance < distance_min){
            distance_min = distance;
            node_proche=*inode;
        }
    }
    return node_proche;</pre>
```

Changement de sous-domaines et sortie du domaine de simulation :

```
void IpgModule::updateParticleCell()
 // pour le parallèle
 IParallelMng * const pm = subDomain()->parallelMng();
 Int32UniqueArray particles to send;
 Int32UniqueArray subdomain to send;
 ENUMERATE PARTICLE (ipart, m activeParticlesGroup) {
   // on récupère les coordonnées de la particule et de sa cellule
   // si ce n'est pas le cas, on parcourt toutes les cellules pour lui associer la bonne
   if (!(isCoordInCell(particule coord, nodes coord))){
     // on doit également vérifier que la cellule n'est pas sortie du domaine
     bool particle is out = 1;
     ENUMERATE CELL ( icell, allCells() ) {
       // récupération de la cellule et de ses noeuds
       // si la particule est géométriquement dans la cellule, on associe les deux
       if (isCoordInCell(particule coord, nodes coord new)){
         m particles family->setParticleCell(particle, cell);
         particle is out=0;
         break;
     // Quand on arrive ici, la particule peut être dans trois états :
     // 2 - sa nouvelle cellule est une maille fantôme
     // 3 - la particule n'a pas de nouvelle cellule (particle is out == 1)
     // cas 1 : on est content, on ne fait rien de plus
     // cas 2 : on doit envoyer la particule dans son nouveau sous-domaine
     // cas 3 : on supprime la particule (et tant pis pour le cas 3 (b))
```

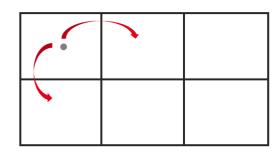


```
if (!ipart->cell().isOwn() && !particle is out){
      particles to send.add( ipart.localId() );
      subdomain to send.add( ipart->cell().owner() );
    UniqueArray<Int32> particle to remove;
    if (particle is out){
      particle to remove.add(ipart.localId()):
      m activeParticlesGroup.removeItems(particle to remove);
// cas 2 : il faut faire l'échange de particules entre sous-domaines
if (pm->isParallel()){
  Int32ConstArrayView send particles(particles to send);
  Int32ConstArrayView send subdomain(subdomain to send);
 IParticleExchanger *pe = getParticleExchanger();
  pe->initialize(m item family);
  pe->beginNewExchange(send particles.size());
  Int32UniqueArray particles to receive;
  pe->exchangeItems(send particles.size(),
                    send particles.
                    send subdomain,
                    &particles to receive,
  m activeParticlesGroup.addItems(particles to receive);
```



Changement de sous-domaines et sortie du domaine de simulation :

```
ENUMERATE PARTICLE (ipart, m activeParticlesGroup) -
   ENUMERATE CELL ( icell, allCells() )
     if (isCoordInCell(particule coord, nodes coord new)){
       m particles family->setParticleCell(particle, cell);
 MaHyGo2-: Journée Arcane la particule dans son nouveau sous-domaine
```



```
if (!ipart->cell().isOwn() && !particle is out){
    particles to send.add( ipart.localId() );
    subdomain to send.add( ipart->cell().owner() );
    particle to remove.add(ipart.localId());
   m activeParticlesGroup.removeItems(particle to remove);
IParticleExchanger *pe = getParticleExchanger();
pe->initialize(m item family);
pe->beginNewExchange(send particles.size());
Int32UniqueArray particles to receive;
pe->exchangeItems(send particles.size(),
                  send particles,
                  send subdomain,
                  &particles to receive,
m activeParticlesGroup.addItems(particles to receive);
```

Initialisation et localisation des particules à partir d'un fichier A_1

Localisation d'un point dans un quadrilatère :

- Méthode de ray-casting.
- Nombre d'intersections rayon/face : pair = en-dehors ; impair = en-dedans

Influence du Service BasicParticlesExchanger (MPI) sur l'architecture du code :

- Échange de particules entre sous-domaines voisins (non précisé dans la doc)
- Impossible de lire tout le fichier sur le processus 0 puis d'envoyer directement toutes les particules vers le bon sous-domaine
- Lecture du fichier sur tous les processus : attention à l'Uniqueld

Choix final:

- Utilisation d'un buffer (position et temps d'injection)
 connu de tous les sous domaines + 2 lectures du fichier
- $O(N_{mailles} \times N_{particules})$

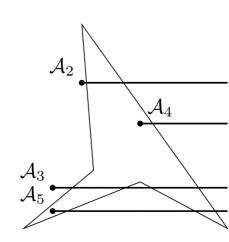
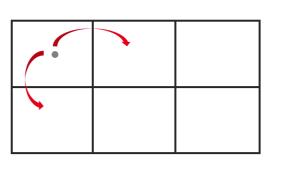
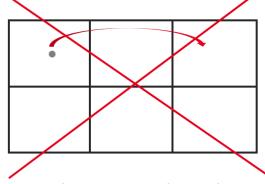


Schéma du ray-casting





Echanges MPI possibles (gauche) et interdits (droite)



Initialisation et localisation des particules à partir d'un fichier

Algorithme d'initialisation des particules : 3 étapes

init

1ère lecture fichier: initialisation buffer (position, temps d'injection) dans chaque sous-domaine

```
m_buffer_coord.add(Real3(xi, yi, zi));
m_buffer_inject_time.add(ti);
```

compute-loop

Parcours du buffer de particules :

Si temps d'injection d'une particule :

Si particule localisée dans une maille appartenant au sous-domaine :

Ajout de la particule à la famille et au groupe.

Initialisation du Uniqueld (= numéro de ligne de la particule dans le fichier)

compute-loop

Parcours des particules du groupe et initialisation de ses variables (2ème lecture du fichier)



Initialisation et localisation des particules à partir

d'un fichier

```
void UserFileInputParticlesService::createParticles()
 // ### si au moins une particule doit être injectée à cette itération
 if ((m t next part >= m global time()-m global deltat()) && (m t next part < m global time())){
         // on les ajoute à la famille et à un groupe
         [...]
         m particles family->addParticles(uids.view(), lids.view());
         m missing data particles group.addItems(lids.view());
         local Id++;
   m particles family->endUpdate(); // should not be forgotten
   // parcours des particules non initialisées
   ENUMERATE PARTICLE (ipart, m missing data particles group) {
       // on initialise les vars de la particule
       [...]
     // on prépare la particule à changer de groupe parce qu'on connait ses vars
     particles to move.add(ipart.localId());
    assignParticleToCell();
   // les particules initialisées deviennent actives
   m missing data particles group.removeItems(particles to move);
   m active particles group.addItems(particles to move);
   m missing data particles group.clear();
   [\ldots]
```

Initialisation et localisation des particules à partir

d'un fichier

```
m particles family->addParticles(uids.view(), lids.view());
     m missing data particles group.addItems(lids.view());
      local Id++;
m particles family->endUpdate(); // should not be forgotten
ENUMERATE PARTICLE (ipart, m missing data particles group) {
// les particules initialisées deviennent actives
m missing data particles group.removeItems(particles to move);
m active particles group.addItems(particles to move);
m missing data particles group.clear();
```

Résultats

Problème d'étude

Une Sphère rigide

Dans un écoulement :

- Uniforme
- Stationnaire
- Sans gravité
- Rétroaction de la particule sur le gaz négligeable

Dans le référentiel du fluide :

$$\frac{\mathrm{d}x_p}{\mathrm{d}t} = u_p$$

$$\frac{\mathrm{d}u_p}{\mathrm{d}t} = -D_p u_p$$

Trainée linéaire :

$$D_p = A$$
, avec $A \in \mathbb{R}$

$$x_p(t) = x_p(0) + \frac{u_p(0)}{A} \left(1 - e^{-At} \right)$$
$$u_p(t) = u_p(0) e^{-At}.$$

Trainée quadratique :

$$D_p = -\frac{3}{8} \frac{\rho C_D}{\rho_p r_p} |u_p|$$

$$x_p(t) = x_p(0) + \frac{1}{A} \ln (1 + A u_p(0) t)$$
$$u_p(t) = \frac{u_p(0)}{1 + A u_p(0) t}$$

$$A = -\frac{3}{8} \frac{\rho C_D}{\rho_p r_p}$$

40

Problème d'étude

Une Sphère rigide

Dans un écoulement :

- Uniforme
- Stationnaire
- Sans gravité
- Rétroaction de la particule sur le gaz négligeable

Dans le référentiel du fluide :

$$\frac{\mathrm{d}x_p}{\mathrm{d}t} = u_p$$

$$\frac{\mathrm{d}u_p}{\mathrm{d}t} = -D_p u_p$$

[9] L. D. Cloutman, 1991

Adimensionnement : $u_{p(0)}$ et r_p

Etat initial:

$$x_p^*(0) = 100$$
 $u_p^*(0) = 0$
 $A^* = 3.85 \times 10^{-5}$

Domaine, maillage et pas de temps :

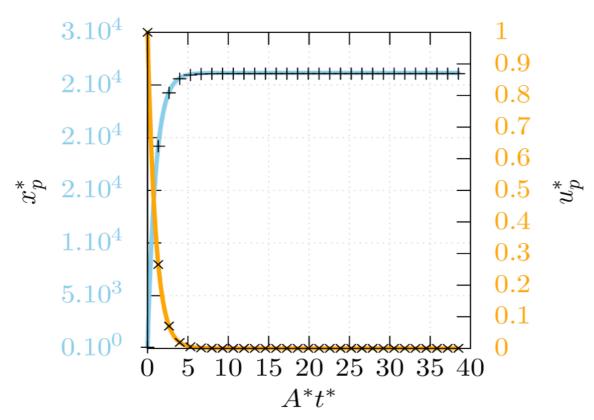
Rectangle
$$L_x^* = 10^5 \times L_y^* = 10^4$$

10 mailles en x, 1 maille en y

$$dt_0^* = 10^{-1}$$
 $CFL = 0.3$ $t_{final}^* = 10^6$

Fluide : gaz parfait, $\gamma = 1.4$

Résultats (modèle Sprays Fins et Sprays Très Fins)



Solution (position et vitesse)

Profils de position et vitesse bien reproduits

→ Les résultats sont quantitativement corrects

Légende :

+ et x : solution analytique

Traits pleins : résultats numériques

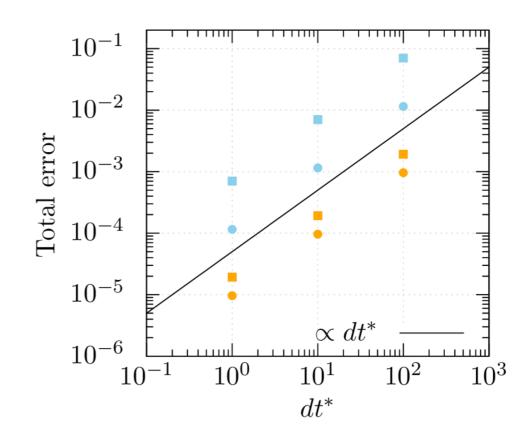


Convergence en dt pour le Service Sprays Très Fins

Erreur totale en norme 2 : $\left(\int_{t_0^*}^{t_{final}^*} (f_{num} - f_{an})^2 dt^*\right)^{1/2}$

Convergence à l'ordre 1 :

- Attendue (traînée : Euler Backward ordre 1)
- Valide l'implémentation



Légende :

Bleu: position Orange: vitesse



Conclusion

MaHyCo, une maquette:

- Développement évolutif en fonction des besoins
- Basée sur la plateforme Arcane (efficacité et performance)
- À terme : résolution de nombreux problèmes physiques (multi matériaux, particules, ...)
- Permet des tests de schémas numériques et de modélisations physiques

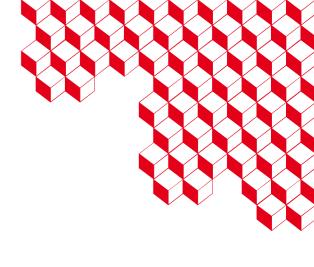
MaHyCo, Arcane et Particules :

- Item PARTICLES : un outil adapté à la simulation d'objets lagrangiens indépendants du maillage
- Gestion similaire (variables, familles et groupe...) aux autres Items Arcane (mailles, ...)
- Quelques subtilités à connaître pour le parallèle MPI
- Validation de l'implémentation des modèles sprays fins et très fins à l'ordre 1
- Portage GPU prévu prochainement









Introduction d'un modèle de spray dans le code MaHyCo

24 mars 2025 – Journée Arcane

Lucas Prouvost

CEA-DIF

Travaux réalisés dans le cadre du LRC MESO pendant l'été 2024

Bibliographie

Schéma hydrodynamique « Lagrange + projection »

- J. von Neumann, R. D. Richtmeyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys. 1950
- M. Wilkins, Use of artificial viscosity in multidimensional fluid dynamic calculations, J. Comp. Phys., 1980
- M. Wilkins, Calculation of elastic-plastic flows, J. Comp. Phys., 1963

Interaction gaz-particules

- A.A. Amsden, P.J. O'Rourke and T.D. Butler, KIVA-II: a computer program for chemically reactive flows with sprays, Report #LA-11560-MS, Los Alamos National Laboratory, 1989.
- J.K. Dukowicz, A particle-fluid numerical model for liquid sprays, J. comp. Phys., 35: 229-253,1980
- L.D. Cloutman, Analytical solutions for the trajectories and thermal histories of unforced particulates, Am. J. Phys., 56: 643-645, 1988
- P. J. O'Rourke, collective drop effects on vaporizing liquid sprays, phd thesis, 1981
- C. Baranger, Modélisation, étude mathématique et simulation des collisions, thèse de doctorat, 2004
- K. D. Kafui, C. Thornton, M. J. Adams, Discrete particle-continuum fluid modelling of gas-solid fluidised beds, chemical engineering science, 2002
- T. B. Anderson and R. Jackson, a fluid mechanical description of fluidized beds, industrial and engineering chemistry fundamentals, 1967
- C. Cercignani, The Boltzmann equation and its applications, Applied Mathematical Science, 1988

MaHyCo – Journée Arcane 13/06/2024 47

MaHyCo: étape Lagrangienne

Équations discrètes

Résolues dans l'ordre suivant :

- Vitesse
- Position nœuds
- Volumes et autres grandeurs géométriques
- Masse/masse volumique
- Viscosité artificielle
- Énergie et pression

$$\boldsymbol{u}_q^{n+1/2} = \boldsymbol{u}_q^{n-1/2} + \frac{dt^n}{m_q^n} \boldsymbol{F}_q^n$$

$$\boldsymbol{x}_q^{n+1} = \boldsymbol{x}_q^n + dt^{n+1/2} \, \boldsymbol{u}_q^{n+1/2}$$

$$m_i^{n+1} = m_i^n$$

$$(\nabla \cdot \boldsymbol{u})_c = \frac{1}{v_c^{n+1/2}} \frac{1}{dt^{n+1/2}} \left(\frac{1}{\rho_c^{n+1}} - \frac{1}{\rho_c^n} \right)$$

$$q_i^{n+1} = \begin{cases} -\frac{c_{il} L_c a_c^n}{v_i^{n+1/2}} (\nabla \cdot \boldsymbol{u})_c + \frac{c_{iq} L_c^2}{v_i^{n+1/2}} ((\nabla \cdot \boldsymbol{u})_c)^2 &, \text{ if } (\nabla \cdot \boldsymbol{u})_c < 0\\ 0 &, \text{ if } (\nabla \cdot \boldsymbol{u})_c \ge 0 \end{cases}$$

MaHyCo: étape Lagrangienne

Énergie interne et pression

Formulation explicite, formellement d'ordre 1

$$e_i^{n+1} = e_i^n - (p_i^n + q_i^{n+1}) \left(\frac{1}{\rho_i^{n+1}} - \frac{1}{\rho_i^n}\right)$$

$$p_{\alpha}^{n+1} = EOS(\rho_{\alpha}^{n+1}, e_{\alpha}^{n+1})$$

Formulation implicite (algorithme de Newton), formellement d'ordre 2

$$e_i^{n+1} = e_i^n - \frac{1}{2} \left(p_i^n + p_i^{n+1} + 2 \, q_i \right) \left(\frac{1}{\rho_i^{n+1}} - \frac{1}{\rho_i^n} \right) \quad p_\alpha^{n+1} = EOS(\rho_\alpha^{n+1}, e_\alpha^{n+1})$$

Cas particulier : gaz parfait, formellement d'ordre 2

$$p = (\gamma - 1)\rho e$$

$$e_i^{n+1} = \frac{A}{B}$$

$$A = e_i^n - \left(\frac{p^n}{2} + q_i\right) \left(\frac{1}{\rho_i^{n+1}} - \frac{1}{\rho_i^n}\right)$$
$$B = 1 + \frac{1}{2}(\gamma - 1)\rho_i^{n+1} \left(\frac{1}{\rho_i^{n+1}} - \frac{1}{\rho_i^n}\right)$$

IPG: Discrétisation choisie

$$\boldsymbol{x}_p^{n+1} - \boldsymbol{x}_p^n = dt^{n+1/2} \, \boldsymbol{u}_p^{n+1/2}$$

$$\hat{\boldsymbol{u}}_p = \boldsymbol{u}_p^{n-1/2} + dt^n \, \boldsymbol{g}$$

$$u_p^{n+1/2} = \frac{\hat{u}_p + dt^n D_p u^{n+1/2}}{1 + dt^n D_p}$$

$$\boldsymbol{u}^{n+1/2} = \left[\boldsymbol{u}^* + \frac{dt^n}{m^n} \sum_{p \in \mathcal{P}(q)} \omega_p \frac{m_p^n D_p}{1 + dt^n D_p} \hat{\boldsymbol{u}}_p \right] \left[1 + \frac{dt^n}{m^n} \sum_{p \in \mathcal{P}(q)} \omega_p \frac{m_p^n D_p}{1 + dt^n D_p} \right]^{-1}$$

Evolution des particules

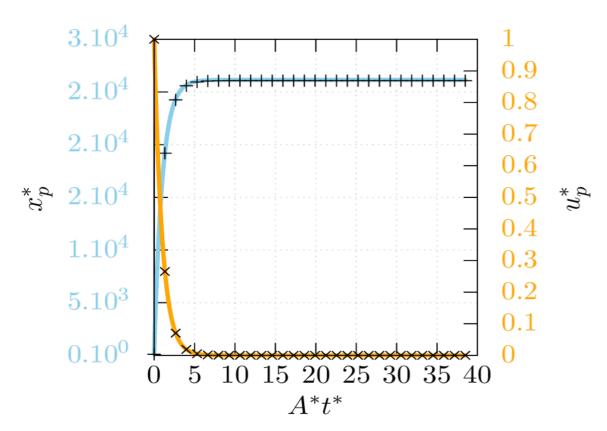
Rétroaction sur le fluide (modification de la vitesse des mailles aux nœuds) :

```
Node SprayFinService::findNodeOfParticle(Particle particule){
    Cell cell = particule.cell();

// la particule appartient au noeud le plus proche
Real distance_min=1e30;
Node node_proche;
for ( NodeEnumerator inode ( cell.nodes() ); inode.hasNext(); ++inode ) {
    Real distance = (m_node_coord[inode] - m_particle_coord[particule]).normL2
    if (distance < distance_min){
        distance_min = distance;
        node_proche=*inode;
    }
}
return node_proche;
}</pre>
```

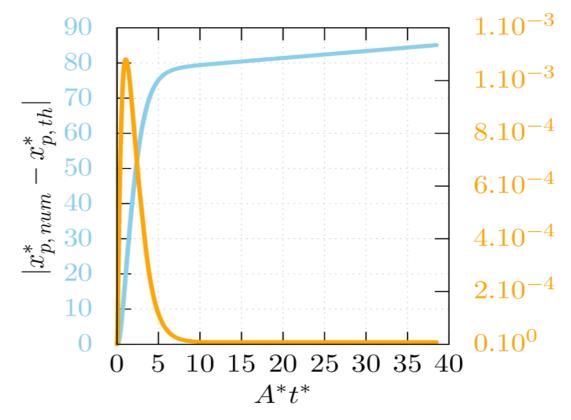
```
void SprayFinService::correctFluidVelocity() {
 const Real dt ( 0.5 * ( m global old deltat() + m global deltat() ) ); // dt^n
 // 0 - Réinitialisation des vecteurs des forces et du denominateur.
 m force.fill ( Real3::zero() );
 m denom.fill ( Real3::zero() );
 ENUMERATE NODE(inode, allNodes()){
   m denom[inode] = 1.;
 ENUMERATE PARTICLE (ipart, activeParticlesGroup) {
   Particle particle = *ipart;
   // 1 - chercher à quel noeud (i.e. cellule duale) appartient la particule
          pour le moment, on considère que la particule appartient au noeud le plus proche
   Node node proche = findNodeOfParticle(particle);
   // 2 - calcul de la force de traînée aux noeuds
   Real Np=m particle weight[ipart];
   Real mp=4./3.*Pi*pow(m particle radius[ipart], 3.)*m particle density[ipart];
   Real3 up chapo = m particle velocity[ipart] + dt*options()->getGravity();
   Real3 Dp = computeDp(particle, node proche);
   Real3 one={1., 1., 1.};
   m denom[node proche] += dt/m node mass[node proche]*Np*(mp*Dp)/(one + dt*Dp);
   m force[node proche] += dt/m node mass[node proche]*Np*(mp*Dp)/(one + dt*Dp)*up chapo;
 ENUMERATE NODE(inode, allNodes()){
   m velocitv[inode] += m force[inode];
   m velocity[inode] /= m denom[inode];
```

Résultats (modèle Sprays Fins et Sprays Très Fins)



Solution (position et vitesse)





Erreur sur la position et la vitesse

Erreur relative < 0,5 % sur la position



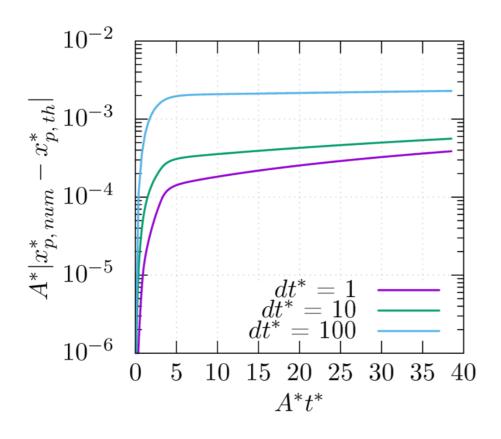
+ et x : solution analytique

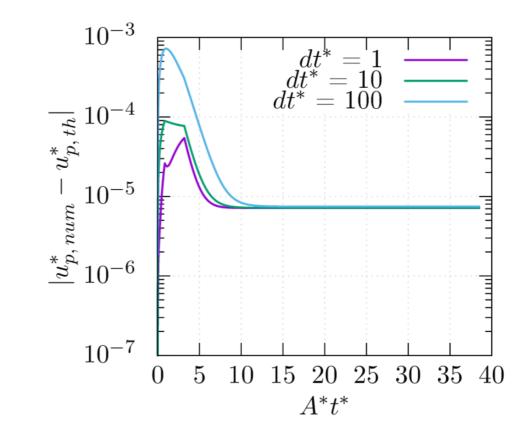
Traits pleins : résultats numériques





Convergence en dt pour le Service Sprays Fins



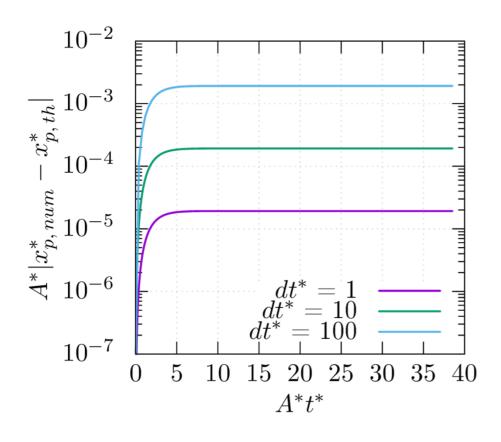


Erreur sur la position : diverge et diminue avec dt mais lentement

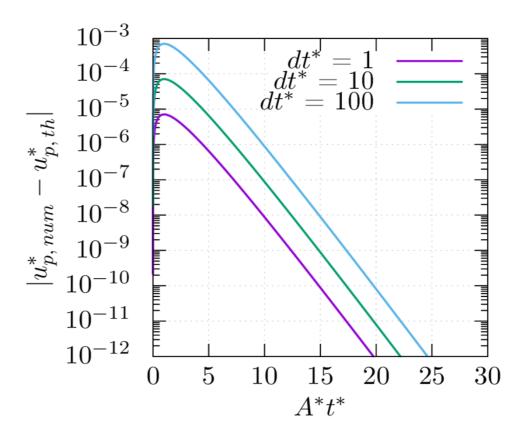
Erreur sur la vitesse : converge et ne diminue pas avec dt

C'est normal : la rétroaction des particules sur le fluide n'est pas négligée pour les sprays fins

Convergence en dt pour le Service Sprays Très Fins



MaHvCo - Journée Arcane



Erreur sur la position : converge et diminue avec dt Erreur sur la vitesse : tend vers 0 et diminue avec dt

C'est normal : le Service Très Fins donne des résultats cohérents avec la théorie (pas de rétroaction) ! Ouf !

Etat des lieux

Travaux réalisés:

- Création et initialisation de particules à partir d'un fichier utilisateur
- Localisation des particules
- Équations des particules
- Couplage gaz particules (sprays fins et très fins, similaire à [9], ordre 1 (Euler backward pour la rétroaction des particules))
- Visualisation des particules (sortie vtk)
- Parallélisation MPI

Travaux en cours:

Mise au propre des cas-tests pour non-régression

Travaux prévus à court terme :

- Couplage gaz particules (sprays épais)
- Initialisation de particules à partir de lois statistiques



ordre 2 : pas simple à obtenir pour une traînée quadratique