

---

## PeliGRIFF

Au cours de ces trois années de thèse, l'étude numérique des écoulements fluide-particules a été réalisée à l'aide des codes PeliGRIFF<sup>1,2</sup> pour la partie fluide et GRAINS3D pour la partie granulaire. Ces logiciels de simulation sont développés au sein d'IFPEN depuis une dizaine d'années et la principale contribution de ce travail de thèse est l'implémentation d'un modèle Euler-Lagrange dans cette plateforme logicielle. Nous présentons ci-après l'architecture générale de ces deux codes ainsi que le schéma de communication leur permettant d'échanger des informations.

### III – 1.1 Généralités

La figure III.1 illustre schématiquement l'imbrication des bibliothèques C++ du code PeliGRIFF. Pour les simulations de suspensions, les trajectoires et les contacts sont traités à l'aide du code de simulation granulaire GRAINS3D. La résolution du problème fluide est quant à elle basée sur la plateforme open-source PELICANS, développée par l'Institut de Radioprotection et de Sécurité Nucléaire (IRSN), dans laquelle une branche spécifique pour la résolution par volumes finis sur grille décalée a été développée par l'équipe PeliGRIFF.

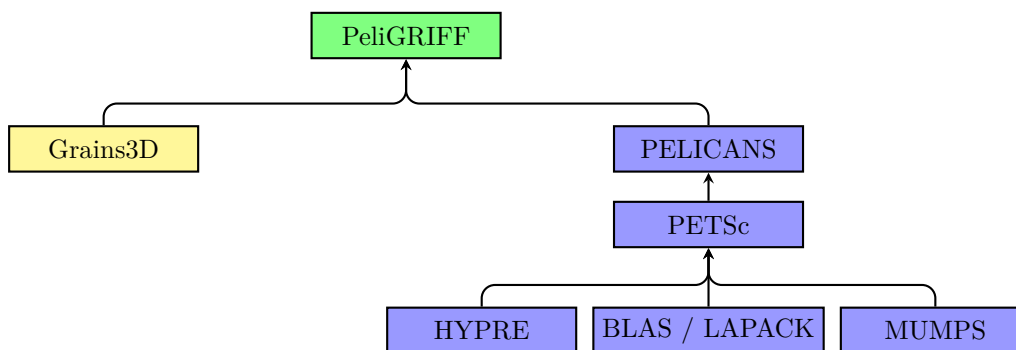


FIGURE III.1 – Diagramme de construction de la plateforme PeliGRIFF

Cette plateforme permet d'utiliser différentes bibliothèques d'algèbre linéaire, à savoir soit des solveurs propres à PELICANS, soit des bibliothèques externes telles que PETSc<sup>3</sup> ou UMFPACK<sup>4</sup>. La bibliothèque de résolution des systèmes linéaires choisie pour PELICANS dans cette étude est PETSc, développée par l'Argonne National Laboratory près de Chicago. La bibliothèque PETSc utilise par défaut les routines provenant des bibliothèques BLAS<sup>5</sup> et LAPACK<sup>6</sup>, mais pour améliorer la rapidité d'inversion des systèmes matriciels, différentes bibliothèques externes peuvent être utilisées. Dans PeliGRIFF, nous utilisons en particulier le préconditionneur multi-grille HYPRE développé au Lawrence Livermore National Laboratory, ainsi que le solveur multifrontal direct MUMPS<sup>7</sup> développé par un regroupement de laboratoires français. Le choix de ces bibliothèques ainsi que des versions utilisées se fait au moment de la compilation du code, par passage d'argument au compilateur. Le compilateur utilisé dans cette étude est gcc 4.8.2

### III – 1.2 Détails

PeliGRIFF est un code modulaire permettant de simuler aussi bien des écoulements bi-fluides à l'aide d'une méthode Level-Set tout en tenant compte de la coalescence de bulles, que les écoulements fluide-particules en prenant en compte ou non les aspects thermiques et/ou chimiques. Le modèle

---

1. Parallel Efficient Library for GRains In Fluid Flow (PeliGRIFF)
2. <http://www.peligriff.com>
3. Portable, Extensible Toolkit for Scientific Computation (PETSc)
4. Unsymmetric Multifrontal sparse Factorization PACKage (UMFPACK)
5. Basic Linear Algebra Subprograms (BLAS)
6. Linear Algebra PACKage (LAPACK)
7. Multifrontal Massively Parallel sparse direct Solver (MUMPS)

DEM-CFD développé au cours de cette thèse à été implémenté dans la plateforme existante dans une version Volumes Finis structurés (voir figure III.2).

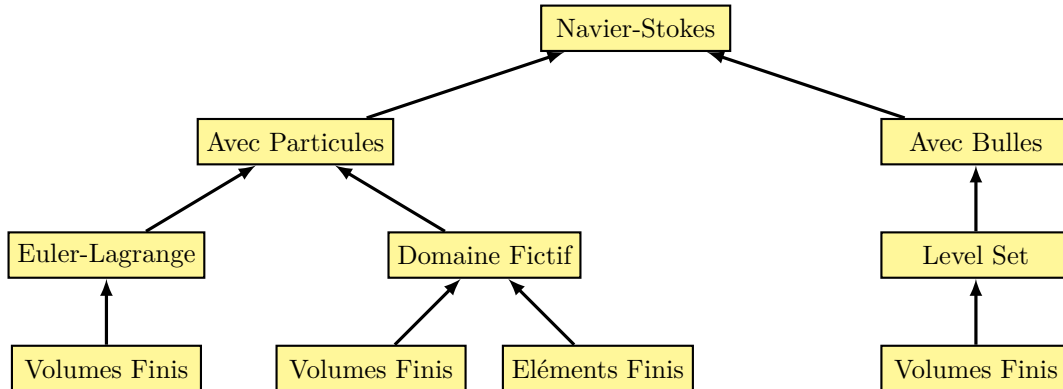


FIGURE III.2 – Structure des différents modules de la plateforme PeliGRIFF

Afin de coupler les solveurs fluide et granulaire, il est nécessaire que ceux-ci s'échangent régulièrement des informations. Pour cela le code PeliGRIFF, qui pilote la simulation, peut accéder librement aux objets de GRAINS3D et en particulier aux particules. Il peut ainsi leur attribuer les données provenant du fluide interpolées au centre de gravité de chacune d'entre elles, comme par exemple le gradient de pression, la vitesse du fluide et fraction volumique de fluide. Le code GRAINS3D peut alors calculer la force de traînée en fonction des données qui viennent de lui être transmises. Après avoir calculé les forces de contacts, le solveur granulaire résout alors le PFD et met à jour la vitesse et la position des particules. PeliGRIFF récupère ensuite la position, le volume et la force de traînée de chaque particule pour mettre à jour la fraction volumique du fluide ainsi que pour calculer la force volumique résultant de l'action des particules sur le fluide (voir équation (II.115)).

La figure III.3 illustre schématiquement l'échange d'informations entre les deux codes. Une version plus détaillée de l'enchaînement des opérations réalisées est présentée sous forme d'algorithme en annexe C.

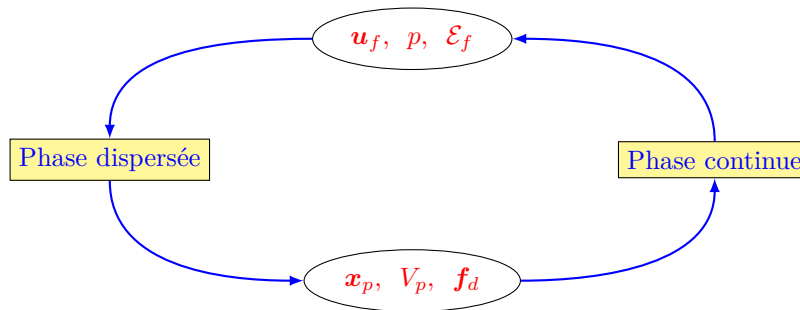


FIGURE III.3 – Couplage et échange d'informations entre les 2 phases

Les codes GRAINS3D et PeliGRIFF sont entièrement parallélisés par une méthode de décomposition de domaine à l'aide de la librairie Message Passing Interface (MPI). Le domaine d'étude est ainsi décomposé en plusieurs sous-domaines qui sont pris en charge par différentes unités de calcul, également appelé cœur de calcul. La décomposition de domaine utilisée dans cette étude est la même pour les parties fluide et solide. Ainsi, la communication entre les deux codes est optimisée car les informations des particules sont directement accessibles (sous forme de pointeurs) par le fluide. Certaines implémentations utilisent quand à elles des décompositions de domaine différentes pour les phases fluide et solide, à savoir généralement un seul processeur pour la phase fluide et plusieurs pour la phase granulaire. Les informations granulaires provenant des différents sous-domaines sont alors envoyées au processeur traitant le fluide sous forme de message MPI, ce processeur renvoie ensuite les informations du fluide à chacun des sous-domaines traitant les particules. Ce schéma de communication ne semble

pas optimum car une grande quantité d'informations est envoyée à chaque itération, ce qui engendre du temps de communication qu'il est préférable de réduire. Les performances parallèles et les temps de calculs des différentes parties du codes sont présentés plus loin dans ce chapitre (voir §III – 6). Pour plus d'informations concernant ces deux codes et les méthodes numériques qui y sont implémentées, voir [224, 222, 223, 237, 236].

## III – 2 Modèle implémenté

Comme nous l'avons détaillé précédemment (voir §II – 2.1), l'écoulement fluide est décrit par les équations de Navier-Stokes :

$$\begin{cases} \frac{\partial \mathcal{E}_f}{\partial t} + \nabla \cdot (\mathcal{E}_f \mathbf{u}_f) = 0 & \text{(III.1a)} \\ \rho_f \left( \frac{\partial (\mathcal{E}_f \mathbf{u}_f)}{\partial t} + \nabla \cdot (\mathcal{E}_f \mathbf{u}_f \mathbf{u}_f) \right) = -\nabla p - \mathbf{F}_{pf} + \nabla \cdot (2\mu \mathcal{E}_f \mathbf{D}) & \text{(III.1b)} \\ \mathbf{F}_{pf} = \frac{\sum_{i=1}^{n_p} \mathbf{f}_{fp,i}}{\Omega} & \text{(III.1c)} \end{cases}$$

La trajectoire d'une particule  $i$  est quant à elle obtenue en résolvant le système d'équations de Newton :

$$\begin{cases} m_p \frac{d\mathbf{v}_p}{dt} = V_p (\rho_p - \rho_f) \mathbf{g} + \mathbf{f}_c + \mathbf{f}_d & \text{(III.2a)} \\ \mathbf{I}_p \frac{d\omega_p}{dt} = \mathbf{T}_p & \text{(III.2b)} \end{cases}$$

A la différence de l'équation de QDM du système (II.5), l'équation (III.1b) ne comprend pas le terme de gravité  $\rho_f \mathcal{E}_f \mathbf{g}$ . La pression ainsi calculée correspond à la pression dynamique et ne comprend pas la pression hydrostatique correspondant à la colonne de fluide. Néanmoins, en prenant en compte le poids relatif dans l'équation de Newton (III.2), la poussée d'Archimède subit par les particules est correcte.

## III – 3 Outils numériques

Afin de résoudre les équations aux dérivées partielles du système (III.1), celles-ci sont discrétisées spatialement et temporellement.

### III – 3.1 Discrétisation spatiale

Dans cette étude, nous ne traitons que des géométries simples, parallélépipédiques rectangles pour la plupart, que nous discrétisons à l'aide de maillages cartésiens qui peuvent être uniformes ou non. Le domaine d'étude est discrétisé à l'aide d'une grille décalée et le système (III.1) est résolu à l'aide d'un schéma volumes finis Marker And Cell (MAC) [93]. Comme l'illustre la figure III.4, la pression  $p$  et la fraction volumique de fluide  $\mathcal{E}_f$  sont localisées au centre des mailles, et les vitesses aux interfaces des volumes de contrôle.

### III – 3.2 Ecriture des termes

Nous présentons ci-après le détail de la discrétisation spatiale des termes de diffusion et d'advection du système (III.1).

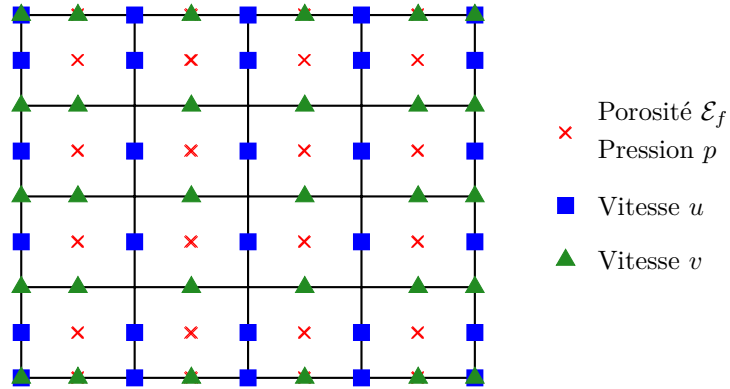


FIGURE III.4 – Positions des variables sur la grille décalée en 2D

### III – 3.2.1 Terme de diffusion

En deux dimensions, le terme de diffusion de l'équation (III.1b) peut s'écrire :

$$\nabla \cdot (2\mu\mathcal{E}_f \mathbf{D}) = 2 \left( \begin{aligned} &\frac{\partial}{\partial x} \left( \mu\mathcal{E}_f \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\mu\mathcal{E}_f}{2} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \\ &\frac{\partial}{\partial x} \left( \frac{\mu\mathcal{E}_f}{2} \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right) + \frac{\partial}{\partial y} \left( \mu\mathcal{E}_f \frac{\partial v}{\partial y} \right) \end{aligned} \right) \quad (\text{III.3})$$

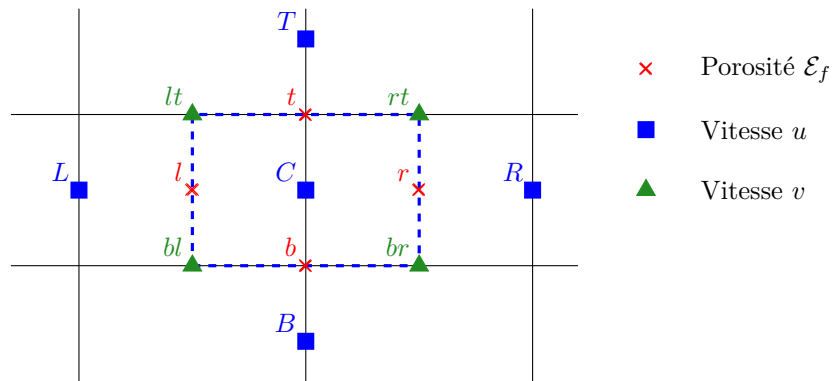


FIGURE III.5 – Cellule de contrôle et position des variables pour le calcul de la première composante du terme de diffusion

En intégrant la première composante sur un volume de contrôle  $d\Omega_u$  de la grille décalée (voir figure

III.5), nous pouvons écrire la formulation continue suivante :

$$\begin{aligned}
\nabla_x \cdot (2\mu\mathcal{E}_f\mathbf{D}) &= \int_{V_u} \left( \frac{\partial}{\partial x} \left( \mu\mathcal{E}_f \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\mu\mathcal{E}_f}{2} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \right) d\Omega_u \\
&= \oint_{\partial V_u} \mu\mathcal{E}_f \frac{\partial u}{\partial x} \mathbf{n}_x dS_u + \oint_{\partial V_u} \frac{\mu\mathcal{E}_f}{2} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \cdot \mathbf{n}_y dS_u \\
&= \left[ \mu\mathcal{E}_{f,r} \left( \frac{\partial u}{\partial x} \right)_r - \mu\mathcal{E}_{f,l} \left( \frac{\partial u}{\partial x} \right)_l \right] \Delta y_C \\
&\quad + \left[ \frac{\mu\mathcal{E}_{f,t}}{2} \left( \frac{\partial u}{\partial y} \right)_t - \frac{\mu\mathcal{E}_{f,b}}{2} \left( \frac{\partial u}{\partial y} \right)_b + \frac{\mu\mathcal{E}_{f,t}}{2} \left( \frac{\partial v}{\partial x} \right)_t - \frac{\mu\mathcal{E}_{f,b}}{2} \left( \frac{\partial v}{\partial x} \right)_b \right] \Delta x_C
\end{aligned}$$

Que nous écrivons sous la forme discrète :

$$\begin{aligned}
\nabla_x \cdot (2\mu\mathcal{E}_f\mathbf{D}) &\approx \mu\mathcal{E}_{f,r} \frac{\Delta y_C}{\Delta x_r} (u_R - u_C) - \mu\mathcal{E}_{f,l} \frac{\Delta y_C}{\Delta x_l} (u_C - u_L) \\
&\quad + \frac{\mu\mathcal{E}_{f,t}}{2} \frac{\Delta x_C}{\Delta y_t} (u_T - u_C) - \frac{\mu\mathcal{E}_{f,b}}{2} \frac{\Delta x_C}{\Delta y_b} (u_C - u_B) \\
&\quad + \frac{\mu\mathcal{E}_{f,t}}{2} \frac{\Delta x_C}{\Delta x_t} (v_{rt} - v_{lt}) - \frac{\mu\mathcal{E}_{f,b}}{2} \frac{\Delta x_C}{\Delta x_b} (v_{rb} - v_{lb})
\end{aligned}$$

Les fractions volumiques  $\mathcal{E}_{f,t}$  et  $\mathcal{E}_{f,b}$  sont obtenues à l'aide d'une interpolation multilinéaire des nœuds voisins à l'ordre 1. L'écriture de la deuxième composante est similaire à la première et l'extension à la troisième dimension se fait quant à elle naturellement.

### III – 3.2.2 Terme d'advection

Le terme de transport de l'équation de QDM (III.1b) est utilisé sous la forme conservative  $\nabla \cdot (\mathcal{E}_f \mathbf{u}_f \mathbf{u}_f)$ . Ainsi, en intégrant sur un volume de contrôle  $d\Omega \subset \mathbb{R}^n$ , et en appliquant le théorème de Green-Ostrograski<sup>8</sup>, nous obtenons l'intégrale suivante sur la surface de contrôle  $dS \subset \mathbb{R}^{n-1}$  :

$$\int_V (\nabla \cdot (\mathcal{E}_f \mathbf{u}_f \mathbf{u}_f)) \cdot d\Omega = \oint_{\partial V} (\mathcal{E}_f \mathbf{u}_f \mathbf{u}_f \cdot \mathbf{n}) dS \quad (\text{III.4})$$

La quantité advectée est la vitesse superficielle du fluide  $\mathcal{E}_f \mathbf{u}_f$ . Comme les nœuds de vitesse et de fraction volumique ne sont pas localisés aux même endroits (voir §III – 3.1), le champ  $\mathcal{E}_f$  est interpolé à l'emplacement des nœuds de vitesse, il est alors noté  $\tilde{\mathcal{E}}_f$ . Dans la suite de cette partie concernant l'implémentation du terme de transport, cette quantité est définie par la lettre  $\Phi$ . Ainsi le terme d'advection s'écrit  $\nabla \cdot (\Phi \mathbf{u}_f)$ .

Nous présentons ci-après le détail de l'implémentation de l'équation (III.4) en deux dimensions pour la première composante du terme d'advection. Deux schémas numériques ont été implémentés pour la discrétisation du terme de transport, un schéma Upwind d'ordre 1 et un schéma TVD d'ordre 2.

#### III – 3.2.2.a Schéma upwind

La figure III.6 illustre la discrétisation spatiale utilisée pour le transport de la quantité  $\Phi$  selon  $x$ . Les composantes de vitesse  $\tilde{u}_l$ ,  $\tilde{u}_r$ ,  $\tilde{v}_b$  et  $\tilde{v}_t$  sont calculées au centre des faces du volume de contrôle (en vert sur le schéma) à partir d'une interpolation linéaire des nœuds voisins. L'écriture du terme d'advection dans le cas d'un maillage cartésien est particulièrement intuitive car cela revient à un

8. également appelé théorème de flux-divergence

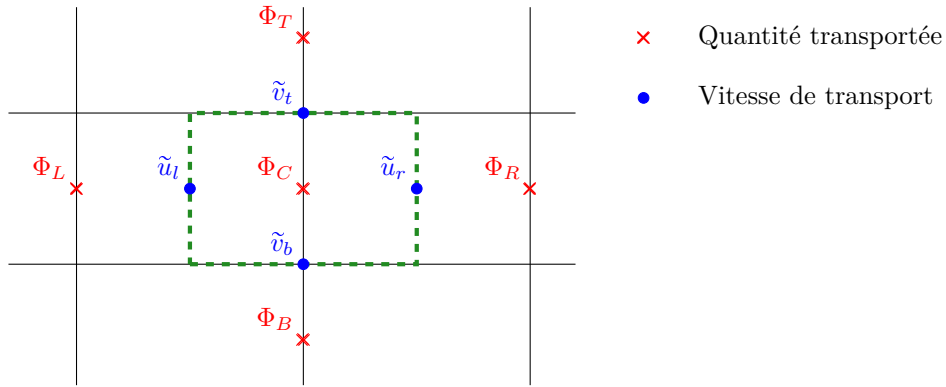


FIGURE III.6 – Cellule de contrôle et position des variables pour le calcul de la première composante du terme de transport avec la méthode upwind

calcul de différences finies dans chaque direction. Ainsi, dans le cas où ces vitesses sont positives, nous pouvons écrire :

$$\oint_{\partial V_u} \Phi \mathbf{u}_f \cdot \mathbf{n} dS_u = (\Phi_C \tilde{u}_r - \Phi_L \tilde{u}_l) \cdot \Delta y + (\Phi_C \tilde{v}_t - \Phi_B \tilde{v}_b) \cdot \Delta x \quad (\text{III.5})$$

### III – 3.2.2.b Schéma TVD

Le terme d'advection peut également être résolu à l'aide d'un schéma garantissant la monotonicité<sup>9</sup>, c'est à dire un schéma respectant la propriété de Variation Totale Décroissante (TVD). Ce schéma est d'ordre 2 dans tout le domaine à l'exception du voisinage des bords du domaine. En effet, le stencil employé pour transporter la valeur  $\Phi = \tilde{\mathcal{E}}_f \mathbf{u}_f$  utilise les deux nœuds amonts. Cela pose donc problème pour la couche de cellules voisines des bords du domaine dans le cas où les conditions aux limites ne sont pas périodiques. Le schéma utilisé est alors le même que le schéma upwind à l'ordre 1 présenté précédemment.

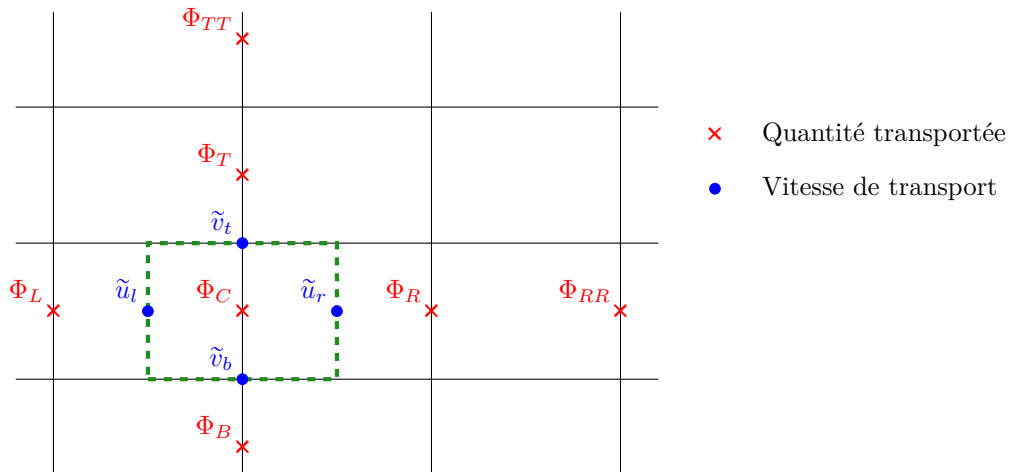


FIGURE III.7 – Cellule de contrôle et position des variables pour le calcul de la première composante du terme de transport avec la méthode TVD

Dans le cas où  $\tilde{u}_l, \tilde{u}_r, \tilde{v}_b$  et  $\tilde{v}_t \geq 0$ , la première composante,  $\Phi = \tilde{\mathcal{E}}_f u_x$  s'écrit sous forme discrète :

$$\int_{\partial S_u} \Phi \mathbf{u}_f \cdot \mathbf{n} dS_u = (F_r - F_l) \cdot \Delta y + (F_t - F_b) \cdot \Delta x \quad (\text{III.6})$$

9. et permettant donc d'éviter les oscillations numériques non désirées

Avec  $F$  le flux sur chacune des faces de la cellule de contrôle. En s'appuyant sur la notation détaillée dans la figure III.7, les flux sont calculés de la façon suivante :

$$\begin{cases} F_r = 0.5 \left[ \tilde{u}_r (\Psi_{RR} + \Psi_R) - |\tilde{u}_r| (\Psi_{RR} - \Psi_R) \right] \\ F_l = 0.5 \left[ \tilde{u}_l (\Psi_L + \Psi_{LL}) - |\tilde{u}_l| (\Psi_L - \Psi_{LL}) \right] \\ F_t = 0.5 \left[ \tilde{u}_t (\Psi_{TT} + \Psi_T) - |\tilde{u}_t| (\Psi_{TT} - \Psi_T) \right] \\ F_b = 0.5 \left[ \tilde{u}_b (\Psi_B + \Psi_{BB}) - |\tilde{u}_b| (\Psi_B - \Psi_{BB}) \right] \end{cases} \quad (\text{III.7})$$

Les termes  $\Psi$  sont des reconstructions linéaires par morceau de la variable discrète  $\Phi$  :

$$\begin{cases} \Psi_{RR} = \Phi_R - \frac{x_R - x_r}{x_{RR} - x_R} \varphi_{rr} (\Phi_{RR} - \Phi_R) \\ \Psi_R = \Phi_C + \frac{x_r - x_C}{x_R - x_C} \varphi_r (\Phi_R - \Phi_C) \\ \Psi_{LL} = \Phi_L + \frac{x_l - x_L}{x_C - x_L} \varphi_l (\Phi_C - \Phi_L) \\ \Psi_L = \Phi_C - \frac{x_C - x_l}{x_R - x_C} \varphi_r (\Phi_R - \Phi_C) \\ \Psi_{TT} = \Phi_T - \frac{y_T - y_t}{y_{TT} - y_T} \varphi_{tt} (\Phi_{TT} - \Phi_T) \\ \Psi_T = \Phi_C + \frac{y_t - y_C}{y_T - y_C} \varphi_t (\Phi_T - \Phi_C) \\ \Psi_{BB} = \Phi_B + \frac{y_b - y_B}{y_C - y_B} \varphi_b (\Phi_C - \Phi_B) \\ \Psi_B = \Phi_C - \frac{y_C - y_b}{y_T - y_C} \varphi_t (\Phi_T - \Phi_C) \end{cases} \quad (\text{III.8})$$

Où  $\varphi$  est un limiteur de pente. Son rôle est de faire repasser localement le schéma au premier ordre afin d'éviter l'apparition d'oscillations non physiques. Dans cette étude, nous utiliserons un limiteur de pente Superbee :

$$\varphi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta)) \quad (\text{III.9})$$

Avec  $\theta$  la pente de la valeur advectée, par exemple :

$$\varphi_{rr} = \varphi(\theta_{rr}) = \varphi\left(\frac{\Phi_R - \Phi_C}{\Phi_{RR} - \Phi_R}\right) \quad (\text{III.10})$$

### III – 4 Résolution numérique

Afin de résoudre numériquement le problème de Navier-Stokes (III.1) dans des régimes d'écoulement ne permettant pas de négliger le terme inertiel de l'équation de QDM, une méthode communément employée consiste à découpler les variables de vitesse et de pression. Nous présentons ci-après le splitting d'opérateurs employé dans cette étude, ainsi que le détail d'un des sous-problèmes émanant de cette séparation d'opérateurs. Dans cette étude, la discrétisation temporelle des équations de conservation de la masse et de la quantité de mouvement est la suivante :

$$\left\{ \begin{array}{l} \rho_f \frac{\mathcal{E}_f^n \mathbf{u}_f^{n+1} - \mathcal{E}_f^{n-1} \mathbf{u}_f^n}{\Delta t} - \nabla \cdot (2\mu \mathcal{E}_f^n \mathbf{D}^{n+1}) = -\nabla p^{n+1} - \mathbf{F}_{pf} - \rho_f \nabla \cdot (\mathcal{E}_f^{n-1} \mathbf{u}_f^n \mathbf{u}_f^n) \end{array} \right. \quad (\text{III.11a})$$

$$\left\{ \begin{array}{l} \frac{\mathcal{E}_f^n - \mathcal{E}_f^{n-1}}{\Delta t} + \nabla \cdot (\mathcal{E}_f^n \mathbf{u}_f^{n+1}) = 0 \end{array} \right. \quad (\text{III.11b})$$

#### III – 4.1 Discrétisation temporelle et splitting d'opérateur

La décomposition d'opérateurs utilisée pour découpler les variables de vitesse et de pression est un schéma à pas fractionnaires de Marchuk-Yanenko (MY) [148, 147]. Le problème initial (III.11) est alors séparé en deux sous-problèmes :

- Un problème d'advection-diffusion

$$\rho_f \frac{\mathcal{E}_f^n \mathbf{u}_f^* - \mathcal{E}_f^{n-1} \mathbf{u}_f^n}{\Delta t} - \nabla \cdot (2\mu \mathcal{E}_f^n \mathbf{D}^*) = -\mathbf{F}_{pf} - \rho_f \nabla \cdot (\mathcal{E}_f^{n-1} \mathbf{u}_f^n \mathbf{u}_f^n) \quad (\text{III.12})$$

- Un problème de Stokes dégénéré

$$\left\{ \begin{array}{l} \rho_f \frac{\mathcal{E}_f^n \mathbf{u}_f^{n+1} - \mathcal{E}_f^n \mathbf{u}_f^*}{\Delta t} = -\nabla p^{n+1} \\ \frac{\mathcal{E}_f^n - \mathcal{E}_f^{n-1}}{\Delta t} + \nabla \cdot (\mathcal{E}_f^n \mathbf{u}_f^{n+1}) = 0 \end{array} \right. \quad (\text{III.13a})$$

$$\left\{ \begin{array}{l} \rho_f \frac{\mathcal{E}_f^n \mathbf{u}_f^{n+1} - \mathcal{E}_f^n \mathbf{u}_f^*}{\Delta t} = -\nabla p^{n+1} \\ \frac{\mathcal{E}_f^n - \mathcal{E}_f^{n-1}}{\Delta t} + \nabla \cdot (\mathcal{E}_f^n \mathbf{u}_f^{n+1}) = 0 \end{array} \right. \quad (\text{III.13b})$$

Cela permet de traiter chaque problème séparément, et donc de choisir une méthode de résolution adaptée à chacun d'entre eux. La vitesse  $\mathbf{u}_f^*$  est une solution intermédiaire, résultant de la résolution du problème d'advection-diffusion.

Le sous-problème d'advection-diffusion (III.12) est résolu à l'aide de la méthode du gradient conjugué préconditionné. Comme la matrice de masse est à diagonale dominante, nous employons un préconditionneur de Jacobi. Le sous-problème de Stokes dégénéré (III.13) est quant à lui résolu par le biais d'une méthode de projection  $L_2$ .

### III – 4.2 Résolution du problème de Stokes dégénéré

Les principales étapes de la résolution du sous-problème (III.13) sont les suivantes :

1. Tout d'abord, l'opérateur de divergence est appliqué à l'équation (III.13a) :

$$-\Delta p^{n+1} = \frac{\rho_f}{\Delta t} \left( \nabla (\mathcal{E}_f^n \mathbf{u}_f^{n+1}) - \nabla (\mathcal{E}_f^n \mathbf{u}_f^*) \right) \quad (\text{III.14})$$

2. L'équation de conservation de la masse (III.13b) est ensuite injectée dans (III.14) :

$$\Delta p^{n+1} = \frac{\rho_f}{\Delta t} \left( \frac{\mathcal{E}_f^n - \mathcal{E}_f^{n-1}}{\Delta t} + \nabla (\mathcal{E}_f^n \mathbf{u}_f^*) \right) \quad (\text{III.15})$$

3. Puis l'équation (III.14) est intégrée sur un volume de contrôle  $d\Omega$  :

$$\int_{\Omega} \Delta p^{n+1} d\Omega = \frac{\rho_f}{\Delta t} \left( \int_V \frac{\mathcal{E}_f^n - \mathcal{E}_f^{n-1}}{\Delta t} d\Omega + \int_{\partial V} (\mathcal{E}_f^n \mathbf{u}_f^*) \mathbf{n} dS \right) \quad (\text{III.16})$$

4. Une fois le champ de pression connu, le nouveau champ de vitesse peut être calculé à partir de l'équation (III.13a) :

$$\frac{\rho_f \mathcal{E}_f^n}{\Delta t} \mathbf{u}_f^{n+1} = -\nabla p^{n+1} + \frac{\rho_f \mathcal{E}_f^n}{\Delta t} \mathbf{u}_f^* \quad (\text{III.17})$$

Comme évoqué dans la partie traitant du calcul de la porosité (§II – 2.4), l'intégration de l'équation (III.16) peut engendrer des oscillations numériques amenant à l'arrêt des simulations si le calcul de  $\mathcal{E}_f$  n'est pas réalisé correctement. En effet, si la prise en compte d'une particule dans un volume de contrôle est faite de façon binaire lorsque une particule quitte ou entre dans une cellule (*ie.* si la "particle center method" est utilisée), la variation temporelle de la fraction volumique de fluide est discontinue. Ainsi, la dérivée temporelle calculée est d'autant plus grande que le pas de temps est petit :

$$\lim_{\Delta t \rightarrow 0} \frac{\mathcal{E}_f^n - \mathcal{E}_f^{n-1}}{\Delta t} = \pm \infty \quad (\text{III.18})$$

Ce phénomène est désormais évité grâce à l'utilisation de la méthode de calcul de  $\mathcal{E}_f$  présentée dans le paragraphe III – 6.4.2.



### III – 4.3 Couplage et ordre de précision temporels

La méthode en temps employée pour résoudre l'équation de QDM est une méthode implicite de Crank Nicolson à l'ordre 2 [37]. Mais la décomposition d'opérateurs de Marchuk Yanenko employée dans cette étude est à l'ordre 1 avec une erreur commise en  $\mathcal{O}(\Delta t/Re)$ , l'ordre en temps de l'algorithme général n'est donc lui aussi que d'ordre 1. La précision temporelle de cette méthode peut être améliorée en utilisant un splitting temporel tel que le splitting de Strang symétrisé [205] qui permet de récupérer une solution à d'ordre 2 en temps. Pour cela, les problèmes -A- d'advection-diffusion (III.12) et -B- de Stokes dégénéré (III.13) doivent être résolus deux fois pour chaque itération temporelle en respectant l'ordre AB puis BA.

Une autre source d'erreur temporelle concerne le couplage entre les phases fluide et solide. En effet, nous avons vu précédemment que les informations du fluide étaient transmises à la phase solide à la fin de chaque itération et qu'entre deux itérations du solveur fluide, plusieurs itérations du solveur granulaire pouvaient être réalisées si  $\Delta t_p < \Delta t_f$ . Dans l'état actuel de l'implémentation du modèle DEM-CFD, les informations prises en compte par la phase solide pour chaque pas de temps  $\Delta t_p$  sont celles provenant de la dernière itération fluide. Or, le pas de temps du solveur fluide est généralement beaucoup plus grand que celui du solveur granulaire car  $\Delta t_f$  nécessite de respecter la Courant-Fredrich-Levy (CFL) pour un maillage relativement large tandis que  $\Delta t_p$  doit être suffisamment précis pour traiter les contacts interparticulaires. Plusieurs itérations solides sont alors réalisées entre chaque itération fluide comme l'illustre la figure III.8.

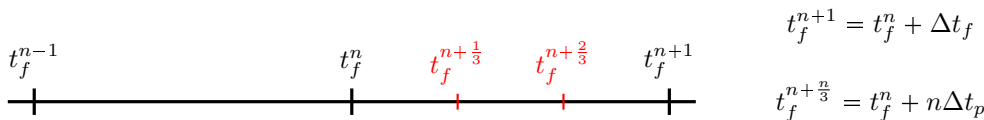


FIGURE III.8 – Illustration des sous-itérations du solveur granulaire entre deux itérations du solveur fluide

Cela a pour conséquence de créer une erreur de couplage temporelle entre les deux phases car si les particules se déplacent trop entre deux itérations du solveur fluide, les informations qui leur ont été attribuée à la fin de la dernière itération fluide ne correspondent plus à leur environnement. Pour réduire l'erreur commise, il serait intéressant de réaliser une interpolation temporelle des valeurs du fluide à chaque itération du solveur granulaire pour chaque mise à jour de la position des particules. Cette méthode se rapprocherait alors de la Méthode des Caractéristiques et l'erreur commise serait de l'ordre l'interpolation réalisée.

## III – 5 Conditions aux limites et initiales

Pour que le problème soit bien posé, il est nécessaire d'imposer des conditions au bord et des conditions initiales pour chaque variable du système. Nous nous placerons dans le cas d'un domaine 2D afin de simplifier les notations et les schémas.

### III – 5.1 Conditions aux limites

Les systèmes étudiés dans cette étude sont de forme rectangulaire et les conditions limites peuvent être de 4 types : entrée, sortie, mur et périodique (voir figure III.9).

Pour les conditions d'entrée et de sortie, les particules rebondissent sur ces faces tandis que le fluide, lui, peut s'écouler. Ce type de condition est imposé afin de conserver la même quantité de particules dans le domaine tout au long de la simulation, et cela même si des particules atteignent le haut du domaine, ce qui en pratique n'arrive que très rarement, voir jamais, de telle sorte que le mélange évolue librement.

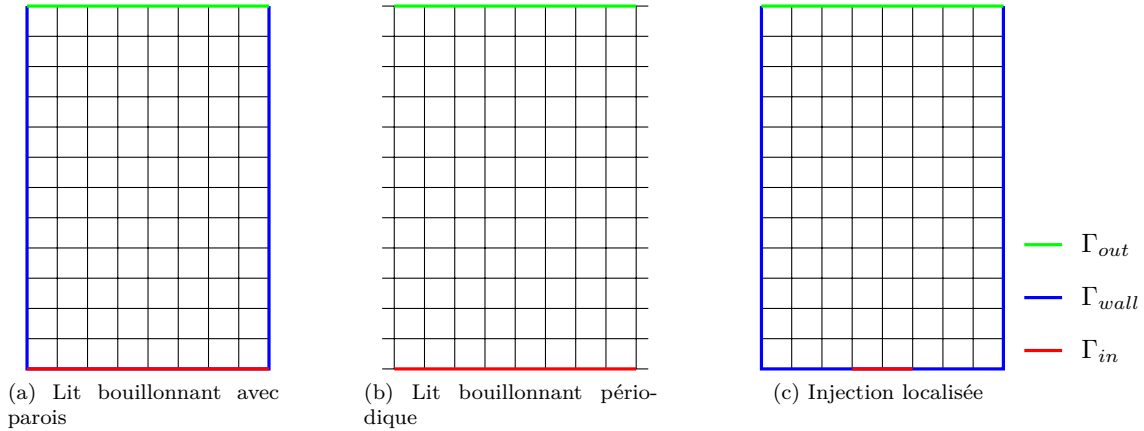


FIGURE III.9 – Différentes conditions aux limites en 2D

### III – 5.1.1 Condition d'entrée

La zone d'entrée du fluide  $\Gamma_{in}$  peut être égale soit à la totalité de la face inférieure du domaine dans le cas d'un lit fluidisé bouillonnant (voir III.9-(a)), soit à une partie de cette même face dans le cas d'un lit pulsé (voir figure III.9-(b)). Sur cette surface d'injection, nous imposons une vitesse d'entrée verticale :

$$\mathbf{U}_{in} = (0, U_{in}), \quad \forall t \geq t_0 \quad (\text{III.19})$$

La condition au bord pour la pression sur la face d'entrée est quant à elle une condition de Neumann. Concernant la condition aux limites de la fraction volumique de fluide, afin de respecter l'équation de conservation de la masse (III.1a) en tout point du domaine, il est nécessaire d'imposer la condition suivante sur la face d'entrée :

$$\mathcal{E}_{f,in} = 1, \quad \forall t \geq t_0 \quad (\text{III.20})$$

En effet pour un état stationnaire, un lit fixe par exemple, l'équation de conservation de la masse dans un des volumes de contrôle jouxtant la face d'entrée se réduit à :

$$\nabla \cdot (\mathcal{E}_f \mathbf{u}_f) = 0 \quad (\text{III.21})$$

Soit, en 2D :

$$(\mathcal{E}_f \mathbf{u}_f)_R - (\mathcal{E}_f \mathbf{u}_f)_L + (\mathcal{E}_f \mathbf{u}_f)_T - (\mathcal{E}_f \mathbf{u}_f)_B = 0 \quad (\text{III.22})$$

Le lit de particules étant homogène, les termes  $(\mathcal{E}_f \mathbf{u}_f)_R$  et  $(\mathcal{E}_f \mathbf{u}_f)_L$  s'annulent. Aussi, la vitesse du fluide en entrée,  $\mathbf{U}_{in}$ , est accélérée dans le lit en raison du rétrécissement de section engendré par les particules, la valeur de cette vitesse interstitielle est  $\mathbf{U}_{in}/\mathcal{E}_f$ . Ainsi, la fraction volumique de fluide sur la face d'entrée doit nécessairement être égale à 1 car  $\mathbf{u}_B = \mathbf{U}_{in}$  :

$$\mathcal{E}_{f,B} = (\mathcal{E}_f \mathbf{u}_f)_T / \mathbf{u}_B \quad (\text{III.23})$$

$$= (\mathcal{E}_{f,bulk} \frac{\mathbf{U}_{in}}{\mathcal{E}_{f,bulk}}) / \mathbf{U}_{in} \quad (\text{III.24})$$

$$= 1 \quad (\text{III.25})$$

L'imposition de cette condition au bord engendre des conséquences non-négligeables sur le résultat de l'interpolation multi-linéaire réalisée pour attribuer la valeur de  $\mathcal{E}_f$  aux particules se trouvant dans la zone comprise entre les nœuds imposés sur la face d'entrée et la première rangée de nœuds calculés.

En effet, comme l'illustre la figure III.10, une particule dont le centre de gravité se trouve sous la première rangée de nœuds  $\mathcal{E}_f$  calculés se verra attribuer une valeur comprise entre la valeur 1 fixée en entrée et la valeur dans le reste du lit (ici  $\mathcal{E}_{f,bulk} = 0.4$  pour l'exemple). En d'autres termes, la valeur

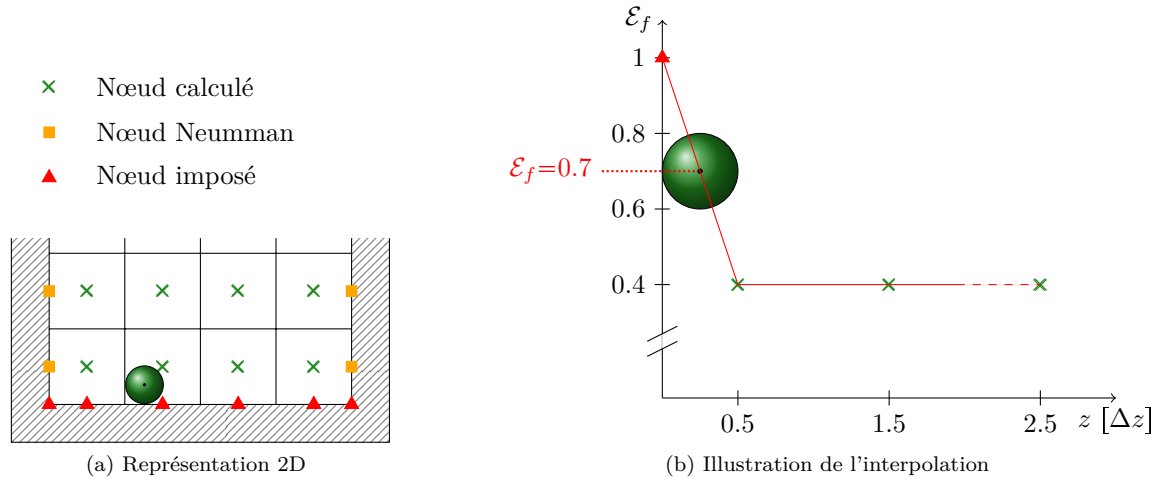


FIGURE III.10 – Interpolation de la fraction volumique de fluide pour une particule proche de la face d'entrée dans le cas où  $\mathcal{E}_f = 1$  est imposée sur la face inférieure du réacteur

$\mathcal{E}_f$  des particules comprises dans cette zone est surestimée. Il en résulte que la force de traînée calculée pour ces particules est plus faible que dans le reste du lit car cette force diminue rapidement lorsque la fraction volumique de fluide augmente (voir figure II.6). Cela a pour conséquence qu'une couche de particules reste présente sur la face inférieure du domaine pour des vitesses d'injection où elles auraient dû être fluidisées. Ce phénomène n'étant pas souhaité, une astuce est employée pour corriger la valeur  $\mathcal{E}_f$  interpolée.

Si l'on ajoute une couche de cellules supplémentaire sous le domaine physique, la valeur  $\mathcal{E}_f$  attribuée à une particule reposant sur la face inférieure du réacteur est alors toujours interpolée entre 1 et la valeur dans le lit, mais cette fois-ci, la position du centre de gravité est plus éloignée du point de pondération de valeur 1 (voir figure III.11). La force de traînée calculée sur les particules présentes dans cet intervalle est ainsi plus proche de celle calculée pour les autres particules du lit, ce qui est plus réaliste.

### III – 5.1.2 Face de sortie

La condition au bord sur la face de sortie est une condition de Neumann pour la vitesse et la fraction volumique de fluide. Remarquons que le flux de fluide sur cette face est constant et égale au flux entrant par le bas du domaine, cela même si le profil de vitesse n'est pas uniforme. La pression sur cette face est quant-à-elle imposée à 0.

Rappelons d'une part que le terme de gravité étant absent de l'équation de quantité de mouvement, la pression dans le système est la pression hydrodynamique. D'autre part, comme nous étudions ici des configurations où le régime d'écoulement est modéré, le terme de pression généré par la friction du fluide sur les parois,  $\Delta p_w$ , est faible devant la pression frictionnelle engendrée par la présence de particules dans le fluide. Ainsi, la pression mesurée entre les faces d'entrée et de sortie est assimilable au saut de pression à travers le lit. Notons que dans le cas d'une configuration bi-périodique, nous avons  $\Delta p_w = 0$ .

### III – 5.1.3 Faces latérales

Deux types de conditions aux bords peuvent être imposées sur les faces latérales du domaine  $\Gamma_{wall}$ . Dans un premier cas, ces faces sont des parois que les particules ne peuvent pas traverser et les collisions mur-particules sont traitées de la même façon que les contacts entre particules. Une condition d'adhérence, *ie.* de Dirichlet homogène, est imposée sur ces murs pour le fluide afin d'assurer l'adhérence du fluide à la paroi. Concernant la pression et la fraction volumique de fluide, les conditions

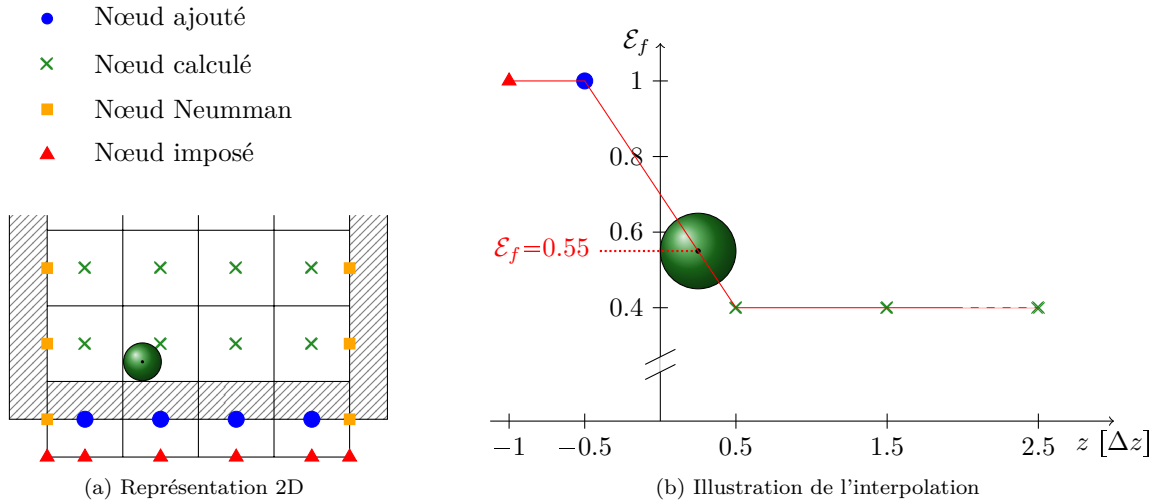


FIGURE III.11 – Interpolation de la fraction volumique de fluide pour une particule proche de la face d'entrée dans le cas où une couche de particules est ajoutée sous la face inférieure du réacteur

aux bords sont de type Neumann homogène. Une telle condition est appliquée pour  $\mathcal{E}_f$  pour éviter le même problème d'interpolation que sur la face d'entrée. Dans le second cas, les conditions aux bords sont périodiques pour la vitesse, la pression, la fraction volumique et les particules. Cela signifie que les nœuds proches des bords du domaine sont en fait vus comme étant voisins des nœuds se trouvant de l'autre côté du domaine (voir figure III.12). De la même façon, une particule se déplaçant en direction d'une frontière latérale apparaîtra de l'autre côté lorsqu'elle la traversera.

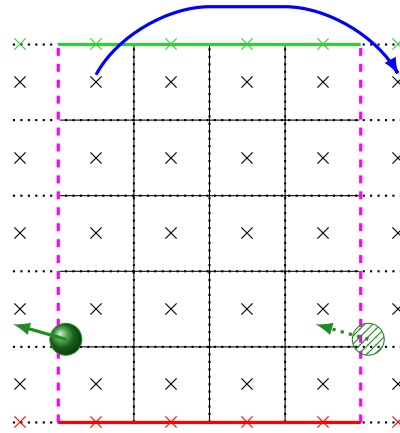


FIGURE III.12 – Conditions aux limites périodiques sur les faces latérales pour les données du fluide comme pour les particules

## III – 5.2 Conditions initiales

### III – 5.2.1 Phase continue

A  $t = 0s$ , la vitesse du fluide est fixée à  $\mathbf{U}^0 = \mathbf{U}_{in}$  dans tout le domaine à l'exception des faces latérales  $\Gamma_{wall}$  où la vitesse est, comme nous l'avons vu dans la sous-section précédente, nulle si ces faces sont des murs. La fraction volumique est quant à elle calculée avant la première itération pour

les nœuds qui ne sont pas contraints, *ie.* tous sauf ceux de la face d'entrée.

A l'état initial, le champ de vitesse  $\mathbf{u}_f^0$  est donc erroné car le fluide ne "voit" pas les particules. De même, lors de la première itération, le champ erroné  $\mathbf{u}_f^0$  est utilisé pour le calcul du champ  $\mathbf{u}_f^1$ , celui-ci est donc lui-même erroné, mais moins qu'à l'état initial puisque la conservation de la masse est vérifiée, et ainsi de suite. La vitesse du fluide dans le lit oscille donc légèrement lors des premières itérations puis ce phénomène s'estompe rapidement. Néanmoins, les effets de cette condition initiale mal posée sont imperceptibles sur la dynamique de l'écoulement.

### III – 5.2.2 Phase dispersée

La position des particules au commencement d'une simulation DEM-CFD peut être déterminée de deux façons différentes. Dans un premier cas, les particules sont placées dans le domaine selon un arrangement choisi par l'utilisateur, qui peut être aléatoire. Cela offre la possibilité de positionner les particules selon des arrangements particuliers dont la porosité est connue de façon analytique. Des exemples de réseaux de particules et leurs caractéristiques sont présentés en annexe A. La seconde méthode permettant d'obtenir la position des particules pour lancer une simulation couplée consiste à réaliser une simulation granulaire sèche, c'est à dire sans couplage avec le fluide, à l'aide du code GRAINS3D et d'utiliser la position des particules issue de cette première simulation. Cela permet par exemple de réaliser une sédimentation granulaire dans le but d'obtenir un lit dense de particules disposées de façon aléatoire. Les valeurs de la porosité du lit ainsi sédimenté peuvent varier entre le loose packing ( $\mathcal{E}_f \simeq 0.42$ ) et le close packing ( $\mathcal{E}_f \simeq 0.36$ ) selon le temps de sédimentation et la vibration ou non du réacteur.

Au cours de cette simulation granulaire sèche, les particules sont soumises à leur propre poids et aux forces de contact présentées dans la partie II – 2.2.1. Néanmoins, la poussée d'Archimède est également prise en compte dans le bilan des forces afin que le recouvrement  $\delta_{ij}$  entre 2 particules  $i$  et  $j$  soit calculé avec le poids relatif des particules tout comme dans une simulation couplée PeliGRIFF. Si ce n'est pas le cas, le lit de particules obtenu numériquement est plus dense car les particules se recouvrent plus. Lors du démarrage d'une simulation couplée, le lit se détend alors brusquement lors de la première itération pour retrouver la situation d'équilibre correspondant à la prise en compte de la poussée d'Archimède dans le PFD (II.13).

## III – 6 Optimisation du temps de calcul

Afin de tester les performances du modèle numérique implémenté, une batterie de simulations a été réalisée sur le super-calculateur ENER110 de IFPEN dont les caractéristiques sont présentées dans le tableau de l'annexe B. Les simulations sont réalisées sur des multiples de 16 cœurs (un nœud plein) afin de garantir une quantité constante de mémoire allouée à chacun des cœurs de calcul et s'affranchir ainsi des problèmes d'accès et de gestion de la mémoire inhérents aux architectures multi-cœurs telles que celle d'ENER110 (voir schéma en annexe B pour plus de renseignements). Les performances ainsi étudiées sont uniquement celles de l'implémentation MPI et non celles liées à la gestion et accès mémoire des données. Nous présentons tout d'abord un aperçu de la répartition du temps de calcul puis les performances parallèles du modèle DEM-CFD sont analysées à travers différents tests de scaling. Enfin, nous détaillons une nouvelle méthode informatique de mise à jour des zones de recouvrement.

### III – 6.1 Répartition du temps de calcul

Les diagrammes de la figure III.13 montrent la répartition des temps de calcul pour une simulation réalisée sur un nœud plein. Pour cette simulation, la charge par cœur est de 30 000 particules et le domaine fluide est composé de  $8 \times 8 \times 80$  cellules. Comme la décomposition de domaine est la même pour les phases fluide et solide, le nombre de cellules fluides traitées par unité de calcul est faible par rapport, par exemple, aux simulations de type DNS pour lesquelles le nombre de cellules par cœur peut atteindre le million d'unités. Pour le modèle DEM-CFD, la majeure partie du temps de calcul est ainsi passée à résoudre le problème granulaire.

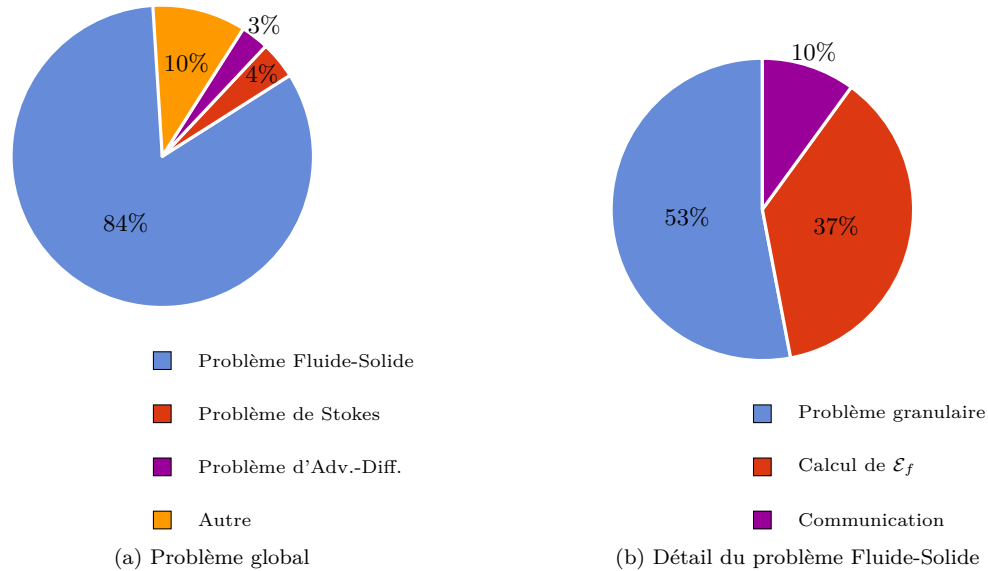


FIGURE III.13 – Répartition du temps de calcul

### III – 6.2 Scaling à charge constante

Des tests de performance parallèle ont également été réalisés pour évaluer la capacité du code à modéliser des systèmes de grande taille. Pour cela, la charge par cœur est maintenue constante, ici 80 000 particules par cœur et des simulations ont été réalisées sur un nombre croissant de nœuds de calculs entre 1 et 32 nœuds. Pour rappel, un nœud de calculs de la machine ENER110 est composé de 16 cœurs. Les tests ont donc été réalisés sur des systèmes comptant entre 1 280 000 et 40 960 000 particules. La figure III.14 montre l'évolution du temps passé dans chacune des parties du code en fonction du nombre de nœuds de calcul utilisé, et donc de la taille du domaine modélisé.

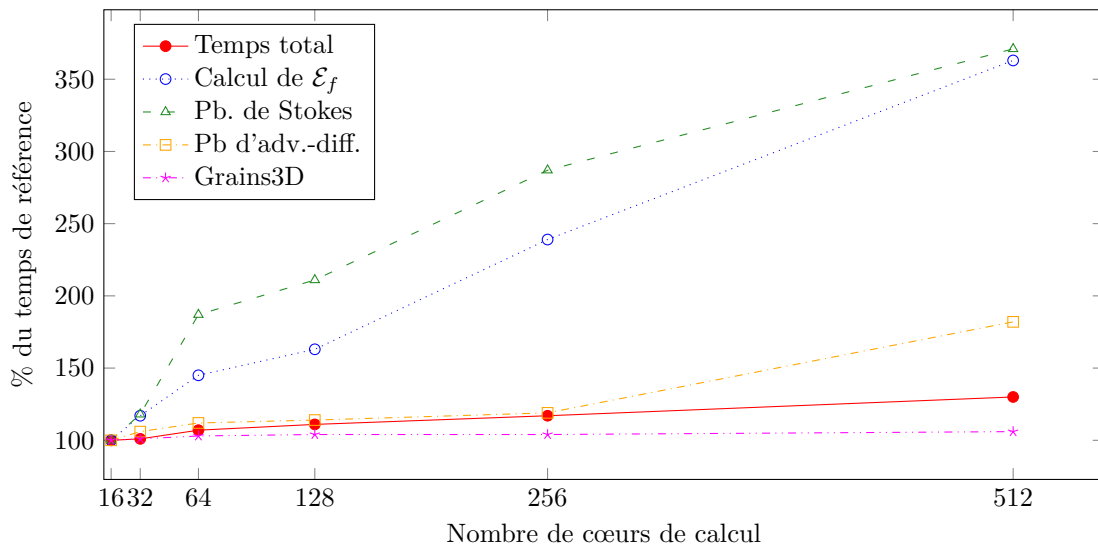


FIGURE III.14 – Scaling à charge constante par nœud. Evolution du temps passé dans les différentes parties du code par rapport au temps passé lors du calcul sur un nœud plein (16 cœurs)

Ce graphique met en évidence le très bon comportement parallèle du solveur granulaire GRAINS3D.

En effet, la résolution du problème inter-particulaire ne prend que 6 % de temps de calcul supplémentaire lors du passage de 16 à 512 cœurs. Il est important de remarquer que comme cette partie du code représente la majeure partie du temps de calcul total, les performances globales du code sont elles aussi relativement bonnes. En effet, le calcul total sur 512 cœurs ne prend que 20% de temps de plus que le cas de référence sur 16 cœurs. En revanche, certaines parties comme la résolution du problème de Stokes et le calcul de  $\mathcal{E}_f$  notamment, présentent un mauvais comportement parallèle car leur temps de calcul augmente considérablement lorsqu'un plus grand nombre de cœurs est utilisé.

Concernant le problème de Stokes, cette chute de performance est probablement due à la méthode de calcul employée pour la résolution du problème de Poisson (III.15). En effet, la matrice du laplacien de pression est initialisée à l'aide du préconditionneur multigrille AMG-BOOMER de la librairie HYPRE. Or cette méthode de résolution est d'autant plus efficace que le nombre de cellules traitées par unité de calcul est grand comme nous le mettons en évidence dans le paragraphe §III – 6.3.2. La mauvaise performance parallèle observée peut également provenir de la décomposition de domaine employée. En effet, le domaine est ici divisé uniquement dans les deux directions transverses et non dans la direction verticale afin d'obtenir une charge de particules par cœur approximativement constante. Les performances du préconditionneur multigrille ne sont donc pas optimales. Une factorisation Incomplète de type "Low-Up" (ILU) a également été testée mais sans amélioration notable.

L'autre partie du code présentant un mauvais comportement parallèle est le calcul de la fraction volumique de fluide. Néanmoins, en étudiant en détail le temps passé dans cette partie du code, il s'avère que la partie chronophage et dont le scaling n'est pas satisfaisant n'est pas le calcul du champ  $\mathcal{E}_f$  à proprement parler mais l'étape qui lui fait suite, à savoir la synchronisation du champ avec les sous-domaines voisins. Une nouvelle implémentation de cette étape de synchronisation a été développée pour améliorer ses performances parallèles, son fonctionnement général est présenté en partie §III – 6.4.

### III – 6.3 Charge par processeur

Afin de parfaire la compréhension du comportement du code, des tests ont également été réalisés en augmentant la charge par cœur de calcul, indépendamment pour le fluide et pour les particules.

#### III – 6.3.1 Augmentation du nombre de particules

Le graphique III.15 représente l'évolution du temps passé par chacune des principales parties du code lorsque le nombre de particules par cœur est multiplié par 2, 4 puis 6, passant alors de 20 000 initialement à 40 000 puis 80 000 et enfin 120 000. Ce test est réalisé sur 1 seul nœud plein et les dimensions du domaine restent identiques, seule la hauteur du lit varie.

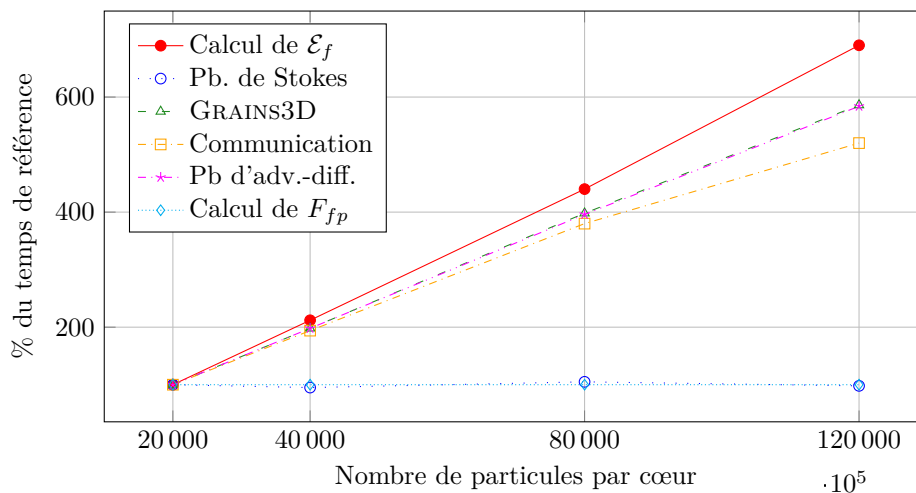


FIGURE III.15 – Evolution du temps de calcul en fonction du nombre de particules par cœur. Charge de référence : 20 000 particules par cœur.

Nous observons tout d'abord que le temps de calcul passé dans la partie granulaire augmente presque linéairement en fonction du nombre de particules par cœur. Cela est en accord avec les performances théoriques de ce code, à savoir une augmentation en  $n \log(n)$  pour un système  $n$  fois plus grand (voir [224] pour plus de détail). Les parties relatives à la transmission des informations de la partie fluide  $(\mathcal{E}_f, \mathbf{u}_f, p)$  vers le module de couplage fluide-solide de GRAINS3D et le calcul de la porosité et de la force  $F_{fp}$  se comportent elles aussi comme on pouvait l'attendre, à savoir une augmentation presque linéaire en fonction du nombre de particules dans le domaine. Concernant la résolution des problèmes d'advection-diffusion (III.12) et de Stokes (III.13), leur durée reste quant à elle inchangée car le nombre de cellules fluides par cœur reste constant. Ce test met donc en évidence le fait qu'une augmentation de la charge de particules par cœur entraîne une augmentation globalement linéaire du temps passé à la résolution du problème granulaire pur et au couplage fluide-particules.

### III – 6.3.2 Augmentation de la taille du domaine fluide

Pour finir cette série de tests sur les performances informatiques, des simulations ont été réalisées sur 16, 32 et 64 cœurs en augmentant progressivement le nombre de cellules fluide par cœur de calcul. La hauteur du domaine est conservée (80 cellules), mais le nombre de cellules sur la section est augmenté, passant de  $8 \times 8$  à  $16 \times 16$  puis à  $32 \times 32$ . Contrairement à une simulation DNS, nous devons ici conserver un certain rapport entre la taille des mailles et le diamètre de particule  $\Delta x/d_p$ , ici cette valeur est de 2.5. Pour augmenter le nombre de cellules sur la section, il est donc nécessaire d'augmenter la taille du domaine. En d'autres termes, il n'est pas possible de raffiner le maillage pour augmenter la charge de cellule par cœur.

A titre informatif, la hauteur initiale du lit de particules est gardée constante mais le nombre de particules sur la section varie en fonction de la taille du domaine modélisé. Le graphique III.16 illustre la progression du temps de calcul par cœur en fonction du nombre de nœuds utilisé pour les trois différentes charges par cœur énoncées précédemment. Cette figure illustre l'amélioration des performances parallèles de la résolution du problème de Stokes en fonction du nombre de cellules fluides par cœur.

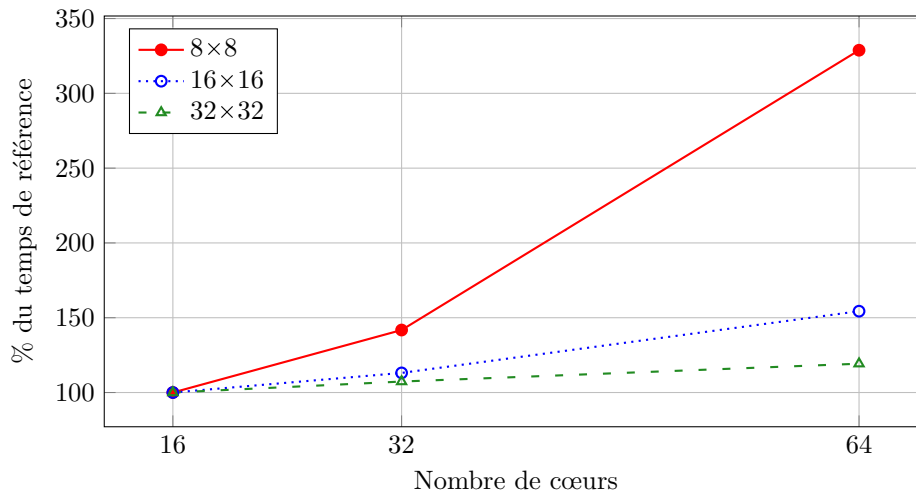


FIGURE III.16 – Scaling de la résolution du problème de Stokes pour différentes charges de cellules fluides par cœur

### III – 6.4 Mise à jour du champ $\mathcal{E}_f$

Les tests de performances parallèles présentés en §III – 6.2 ont mis en évidence un mauvais comportement parallèle de la synchronisation du champ  $\mathcal{E}_f$ . Pour tenter de résoudre ce problème, une nouvelle implémentation de la mise à jour des valeurs se trouvant dans la zone de recouvrement a été développée. L'ancienne méthode est d'abord présentée, puis la nouvelle implémentation est détaillée.



Pour rappel, cette étape de mise à jour est nécessaire car un processeur réalise le calcul des valeurs qui lui appartient, mais les processeurs voisins nécessitent de connaître les valeurs des cellules qui sont contigües à leur sous-domaine car elles interviennent dans les schémas numériques.

### III – 6.4.1 Ancienne méthode

La première méthode de communication implémentée utilisait le schéma de communication représenté en figure III.17. Son déroulement est le suivant :

1. Les valeurs calculées par le sous-domaine **A** (entourées en bleu sur la figure III.17) sont copiées dans le vecteur global.
2. Les valeurs du sous-domaine **A**, y compris les valeurs halozones (provenant de la bufferzone du sous-domaine **B**), sont copiées du vecteur global vers le vecteur local.
3. Les valeurs sont copiées du vecteur local vers le champ, ainsi tous les nœuds, y compris ceux de la zone de recouvrement, ont la bonne valeur.

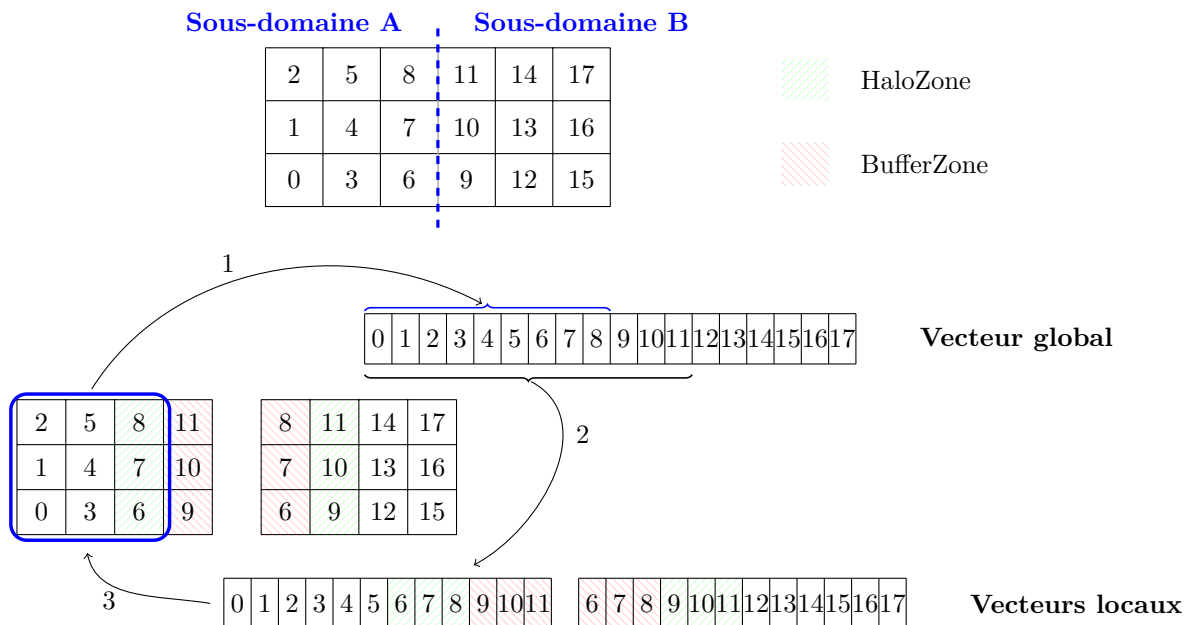


FIGURE III.17 – Ancienne procédure de mise à jour des valeurs de la bande de recouvrement.

Cette méthode, coûteuse en temps de communication, n’est pas optimale car les valeurs ne faisant pas partie de la bufferzone sont tout de même copiées dans le vecteur global puis local avant d’être “recopiées” sur le sous-domaine de départ alors qu’elles ne sont d’aucune utilité aux sous-domaines voisins. Il est à noter que la contre-performance de cette méthode est d’autant plus grande que le nombre de valeurs dans la zone de recouvrement est petit devant le nombre de valeurs dans tout le sous-domaine.

### III – 6.4.2 Nouvelle méthode

Afin d’améliorer les performances de la mise à jour du champ  $\mathcal{E}_f$  dans les zones de recouvrement, une nouvelle méthode de communication a été implémentée. Désormais, le déroulement de l’opération est le suivant :

1. La liste de toutes les valeurs de la bufferzone du sous-domaine 0 (en vert) est constituée.
2. Cette liste est envoyée à tous les sous-domaines voisins.

3. Les voisins identifient les valeurs qui sont dans leur halozone et stockent leur pointeur dans un vecteur.
4. Les voisins envoient ce vecteur au sous-domaine 0.
5. Le sous-domaine 0 associe les valeurs de sa bufferzone aux voisins qui les détiennent dans leur halozone.
6. Par la suite, le sous-domaine 0 n'enverra à ses voisins que les valeurs qui les concernent.

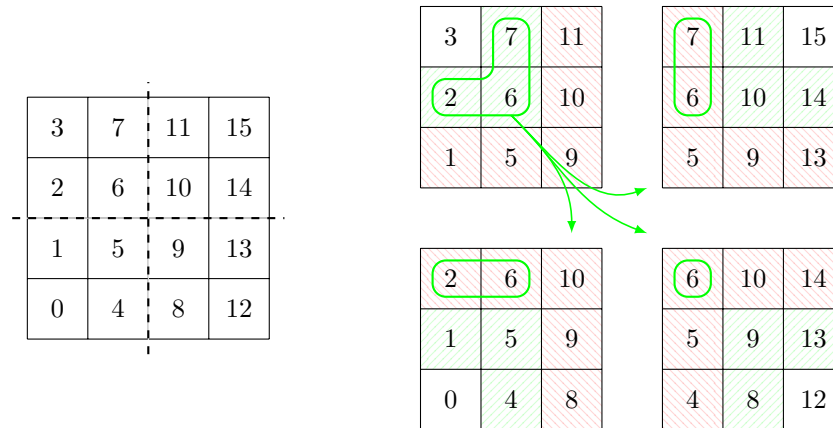


FIGURE III.18 – Nouvelle procédure de mise à jour des valeurs de la bande de recouvrement.

Cette méthode est bien plus efficace que la précédente car la taille des messages envoyés est réduite au strict minimum (seulement les valeurs utiles) et les messages ne sont envoyés qu'aux sous-domaines concernés et non à tous les voisins.

L'implémentation de cette nouvelle méthode de synchronisation du champ  $\mathcal{E}_f$  a permis d'une part de réduire de 10% le temps passé dans cette étape pour un calcul sur un nœud plein, mais surtout, son comportement parallèle a été nettement amélioré puisque cette étape ne prend plus que 30% de plus entre 16 et 256 cœurs, au lieu de 150% avec l'ancienne méthode.

### III – 7 Conclusion

Dans ce troisième chapitre, les méthodes numériques développées et utilisées pour résoudre les équations du modèle DEM-CFD introduit précédemment ont été détaillées. Les performances informatiques de chacune des parties du code ont été étudiées afin d'identifier ses faiblesses et ses points forts. Des solutions ont été proposées et des modifications apportées afin d'améliorer les performances et la précision du modèle implémenté. Les différents tests de scaling, réalisés sur des systèmes comportant jusqu'à plusieurs dizaines de millions de particules, ont abouti à de très bonnes performances parallèles du code. Ainsi nous pouvons affirmer que le modèle DEM-CFD implémenté au cours de cette thèse est un outil permettant de réaliser des simulations de systèmes dont les dimensions, en terme de nombre de particules et de taille de domaine, sont égales à celles de systèmes pouvant également être étudiés à l'aide de modèle Euler-Euler. Des améliorations peuvent encore être apportées au modèle, notamment pour l'interpolation temporelle des données fluides transmises à la phase fluide, ainsi que concernant la distinction entre la grille fluide et les volumes de contrôle employés pour la prise de moyennes spatiales.

