



POLSKA AKADEMIA NAUK  
INSTYTUT MASZYN PRZEPLYWOWYCH

Arch. No. /2016

No. of pages : 41

Classification: external report

Plan position: Contract with MOFPHET-C2 project

No. 8610-5910130261

between EDF R&D<sup>‡</sup> and IMP PAN\*

4<sup>th</sup> part from 01/03/2016 to 01/09/2016

Christophe HENRY, Jacek POZORSKI

**Modelling of agglomeration and deposition of  
colloidal particles carried by a flow:**

**Fourth progress report**

Gdańsk, September 2016

---

\* - Instytut Maszyn Przepływowych, Polska Akademia Nauk

(Institute of Fluid-Flow Machinery, Polish Academy of Sciences)

ul. Fiszera 14, 80-231 Gdańsk, Poland

‡ - EDF R&D - MFEE - I81: Fluid Dynamics, Power Generation and Environment

6 quai Watier, 78401 Chatou, France

# **Modelling of agglomeration and deposition of colloidal particles carried by a flow: Fourth progress report**

---

**Développement et validation de modèles  
dans le module Lagrangian de *Code\_Saturne***

**Development and validation of models  
for the Lagrangian module of *Code\_Saturne***

Christophe HENRY, Jacek POZORSKI

September 1<sup>st</sup>, 2016

# Contents

<b>5 Direct implementations in the Lagrangian module of <i>Code Saturne</i></b>	<b>3</b>
5.1 Modification of the trajectography algorithm . . . . .	4
5.1.1 Theory . . . . .	4
5.1.2 Verification case . . . . .	6
5.2 Adaptation of the multilayer deposition and resuspension models . . . . .	12
5.2.1 Theory . . . . .	12
5.2.2 Case studied . . . . .	12
5.2.3 Conclusion . . . . .	13
5.3 Corrections of the numerical schemes . . . . .	13
5.3.1 Theory . . . . .	13
5.3.2 Verification case . . . . .	14
5.3.3 Validation case . . . . .	26



## Chapter 5

# Direct implementations in the Lagrangian module of *Code Saturne*

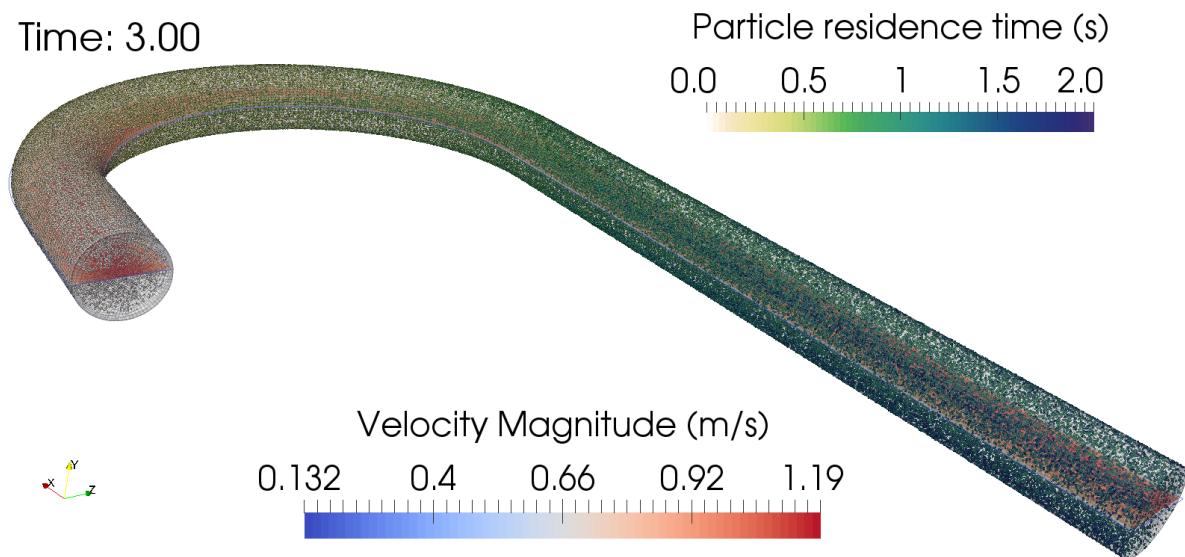


Figure 5.1: Snapshot showing the positions of particles in a U-bend geometry with warped faces.

## 5.1 Modification of the trajectography algorithm

### 5.1.1 Theory

The Lagrangian tracking of particles within a mesh implies that both the global coordinates of each particle and its location within the mesh need to be determined. While the particle displacement is calculated according to the particle equations of motion, a specific trajectography module allows to evaluate the cell to which the particle belongs.

The main idea of the algorithm for trajectography is the following (see also Fig. 5.2): knowing the initial particle location  $O$ , its current cell and the particle location at the end of the time step  $D$ , we check if the particle displacement crosses one of the faces of the cell in which the particle is. For that purpose, for each face of the current cell, we calculate if there is an intersection  $I$  between the particle displacement vector  $\overrightarrow{OD}$  and each subtriangle composing the face (for instance, the face on the right-hand side in Fig. 5.2 can be decomposed into four subtriangles, here  $(GP_1P_2)$ ,  $(GP_2P_3)$ ,  $(GP_3P_4)$  and  $(GP_4P_1)$ ). In this example, to determine whether the intersection with the line  $(OD)$  is inside the triangle  $(GP_1P_2)$ , three conditions need to be fulfilled:

$$\frac{\overrightarrow{OD} \cdot (\overrightarrow{GP_1} \wedge \overrightarrow{GO})}{\overrightarrow{OD} \cdot (\overrightarrow{GP_2} \wedge \overrightarrow{GP_1})} < 0 \quad (5.1a)$$

$$\frac{\overrightarrow{OD} \cdot (\overrightarrow{GP_2} \wedge \overrightarrow{GO})}{\overrightarrow{OD} \cdot (\overrightarrow{GP_2} \wedge \overrightarrow{GP_1})} > 0 \quad (5.1b)$$

$$\frac{\overrightarrow{OD} \cdot (\overrightarrow{P_1P_2} \wedge \overrightarrow{P_1O})}{\overrightarrow{OD} \cdot (\overrightarrow{GP_2} \wedge \overrightarrow{GP_1})} < 0 \quad (5.1c)$$

It should be noted that these three conditions are based on geometric considerations (similar to the Möller-Trumbore algorithm) which amount to checking if the intersection point  $I$  is located on the proper side of each edge of the triangle. For computational purposes, only the sign of these parameters are calculated in the calculation. This choice has been made to avoid to treat properly the case where the intersection is right on the edge (in that case, the calculated value is identical for both faces containing the edge meaning that the intersection exists only on one side of the edge).

If the line  $(OD)$  crosses a triangle, the coordinates of the intersection point  $I$  are calculated using:

$$\mathbf{x}_I = \mathbf{x}_O + t \times \overrightarrow{OD} \quad (5.2)$$

with  $t \in [0, 1[$  given by:

$$t = \frac{\overrightarrow{GO} \cdot (\overrightarrow{GP_2} \wedge \overrightarrow{GP_1})}{\overrightarrow{OD} \cdot (\overrightarrow{GP_2} \wedge \overrightarrow{GP_1})} \quad (5.3)$$

Moreover, Eqs. (5.1) actually provide information whether the whole line  $(OD)$  crosses one

of the faces of the current cell. Comparing the orientation of the displacement vector to the orientation of the local normal to the triangle (oriented towards the cell center) thus gives information on the number of times the line ( $OD$ ) enters (parameter  $n\_in$ ) and leaves (parameter  $n\_out$ ) the current cell. This is used to check if the particle is really inside the current cell, which amounts to the following condition:  $n_{in} = n_{out} > 0$ .

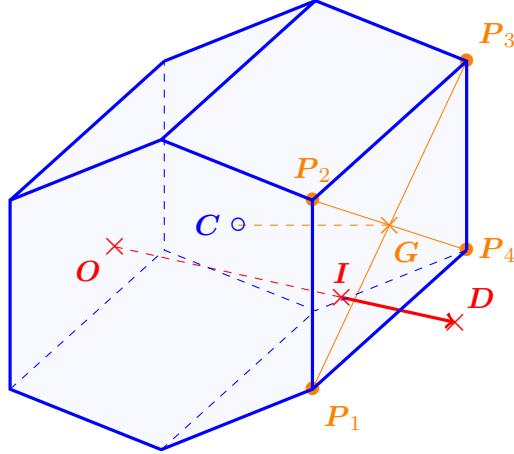


Figure 5.2: Sketch showing a particle displacement from  $O$  to  $D$  within a cell

Besides, a specific treatment has been added to deal with the case of warped faces. As depicted in Fig. 5.3, in this case, the particle displacement vector  $\overrightarrow{OD}$  may cross several subtriangles of a single face. To properly treat such cases, a parameter has been added to measure how many times the displacement vector  $\overrightarrow{OD}$  enters and leaves one face (in a similar way to the one used for the parameters  $n\_in$  and  $n\_out$ ). As a result, depending on the value of this parameter, two different cases can be distinguished: either the particle leaves through the current face (positive parameter) or does not leave the cell through this face (zero value).

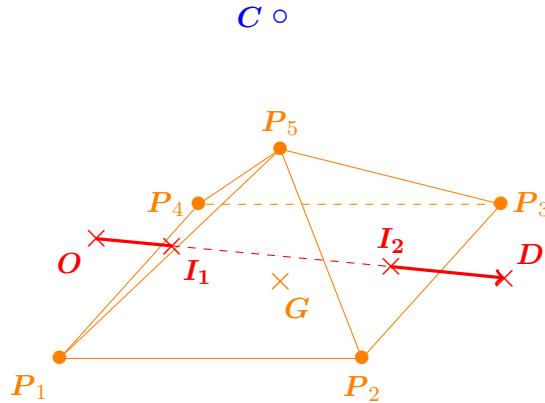


Figure 5.3: Sketch showing a particle displacement from  $O$  to  $D$  going through a warped face

$U_{inlet}$	$\mu_f$	$\rho_f$
$1 \text{ m s}^{-1}$	$1 \times 10^{-3} \text{ kg m}^{-1} \text{s}^{-1}$	$1 \times 10^3 \text{ kg m}^{-3}$

Table 5.1: Fluid properties used in the simulation: inlet velocity  $U_{inlet}$ , Reynolds number  $Re$ , dynamic viscosity  $\mu_f$  and density  $\rho_f$ .

This procedure is repeated until the particles reaches a cell where it remains (no exit through any faces of the new cell). Warning: to avoid high computational costs, a maximum number of 100 cells can be crossed by each particle in a single time step, otherwise it is considered lost (this maximum value can be changed).

### 5.1.2 Verification case

To verify the validity of the present algorithm for particle trajectography, a simulation has been performed in a complex geometry with warped faces.

#### Case description

This test case consists in a flow of water within a U-bend configuration (i.e. a 3D cylindrical pipe with a curve) loaded with colloidal particles.

#### Computational domain

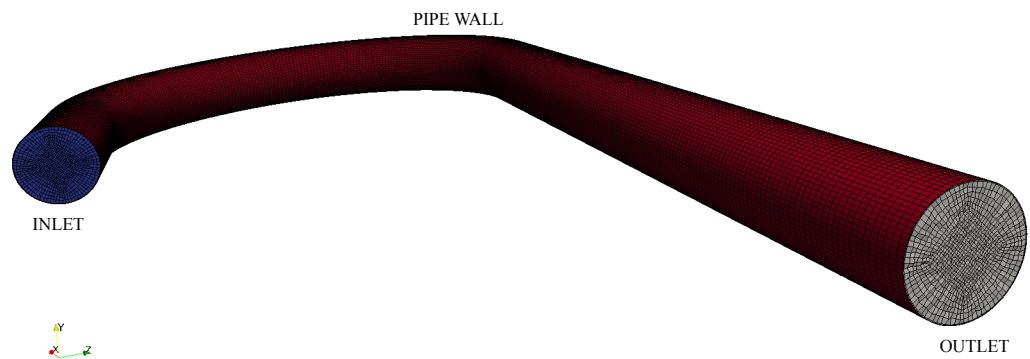
The 3D CFD geometry of the computational domain is shown in Figure (5.4) which displays both the boundary conditions (with inlet, outlet and pipe wall) and a zoom of the mesh structure close to the inlet (where warped faces can be seen). The U-bend cross section is circular with a diameter of 0.0455 m. The ratio between the bend mean radius and the curvature to pipe diameter is  $R_c/D = 3.375$ . The length of the straight pipe section between the outlet and the bend is 0.089 m while the length of the straight pipe section between the bend and the outlet is 0.356 m. The grid has been generated with hexahedral non uniform mesh, which comprises 349 440 cells.

#### Physical model for the fluid flow

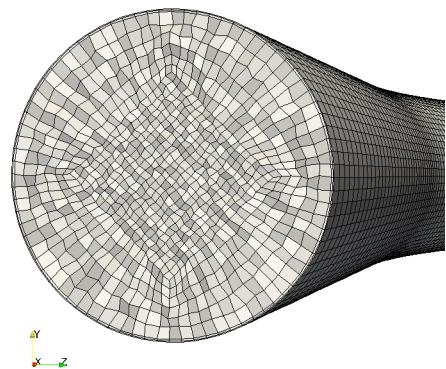
**Generalities** The flow is isothermal and incompressible. Gravity has been neglected. The properties used for the liquid in the flow domain are summarised in Table 5.1:

**Turbulence models** A CFD simulation of the turbulent flow in this U-bend 3D configuration has been performed using *Code\_Saturne*. For that purpose, we have retained:

- Turbulence model:  $k - \varepsilon - PL$  (with default values)



(a) Mesh and boundary conditions



(b) Zoom on the inlet

Figure 5.4: 3D view of the geometry of the computational domain, showing both the boundary conditions (a) and the mesh structure with warped faces close to the inlet (b).

- Unsteady flow

## Initial conditions

The initial conditions are the following:

- reference and initial pressure: 101 300 Pa
- fluid velocity:  $0.0 \text{ m s}^{-1}$
- temperature: 293.15 K

## Boundary conditions

**Inlet boundary** At the inlet, the flow has a uniform profile with the following properties:

- velocity:  $1.0 \text{ m s}^{-1}$
- hydraulic diameter (for turbulence initialisation): 1.0 m

**Wall boundary** The wall boundary condition is non-slip velocity with a smooth wall.

**Outlet boundary** No imposed quantity at the outlet.

## Numerical parameters

The fluid-flow simulation has been performed using the following numerical parameters:

- Constant time step: 0.01 s
- Number of iterations: 100

These parameters ensure the following properties

Field	Minimum	Maximum	Mean
Courant number	0.36	7.9	3.9
Fourier number	0.04	12.1	1.9

Table 5.2: Properties resulting from the parameters used in the simulation (Cournat, Fourier numbers).

## Physical model for particles

**Injection** Particles have been injected in the domain at the inlet. The properties used for the injected particles are:

- Particle diameter:  $1 \times 10^{-5}$  m
- Monodispersed particles
- Particle density:  $2 \times 10^3$  kg m<sup>-3</sup>
- Frequency of injection: 1 (every time step)
- Number of particles in class: 1000

**Boundary condition** The wall boundary conditions for particles have been set to three possible states

- Particles rebound
- Deposition and elimination
- Deposition

**Model for transport** The CFD simulation with the injection of particles has been performed using the Lagrangian module in *Code\_Saturne* and the following properties

- Eulerian-Lagrangian multi-phase treatment: one way coupling
- No additional models associated with the particles
- Turbulent deposition: particle deposition sub-model
- Integration for the stochastic differential equations: first-order scheme
- Particle turbulent dispersion model: activated

**Numerical parameters** The two-phase flow simulation has been performed using the following numerical parameters:

- Calculation restart from the fluid flow simulation
- Constant time step: 0.01 s
- Number of iterations: 300

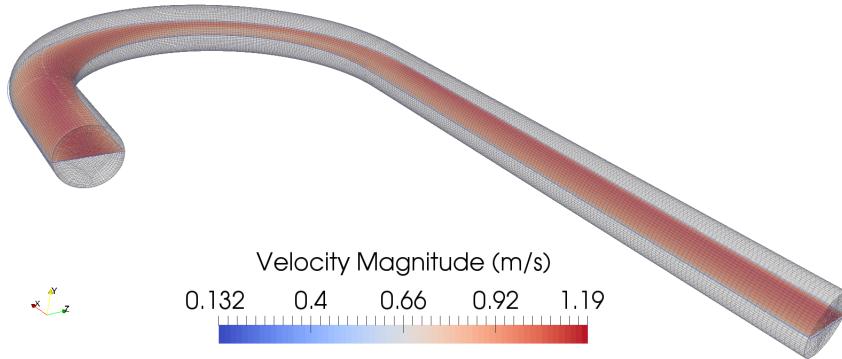


Figure 5.5: Slice of the U-bend geometry showing the fluid velocity obtained from the fluid-flow simulation

## Results

Since the aim of the present verification is to assess the performance of the trajectography algorithm, the fluid flow simulation is very briefly presented. The velocity field inside the U-bend geometry is displayed in Figure 5.5.

Three sets of simulations have then been performed for two-phase flows depending on the wall boundary conditions retained for particles (either rebound, deposition and elimination, deposition). A 3D view of the particles inside the U-bend pipe obtained numerically is shown in Figure 5.6 at different elapsed time: it can be seen that the particles strictly remain within the domain and that no particles are lost.

A comparison of the number of particles within the computational domain as the simulation time increases is show in Figure 5.7a where three different behavior can be seen depending on the wall condition:

- Rebound: in that case, particles do not stick to the surface so the number of particles in the simulation increases with time until it reaches a plateau value (due to the fact that the number of particles entering the domain is compensated by the number of particles exiting through the outlet face).
- Deposition and elimination: the evolution is similar to the case with rebound except that the plateau value reached is smaller by one order of magnitude (due to the fact that the number of particles entering through the inlet is compensated by both the number of particles exiting the domain and the number of particles eliminated due to deposition on the pipe wall).
- Deposition: in that case, it can be seen that the number of particles increases continuously since particles are constantly injected and some of them deposit on the pipe wall (and stick there). This simple analysis also explains the fact that two slopes can be seen for the

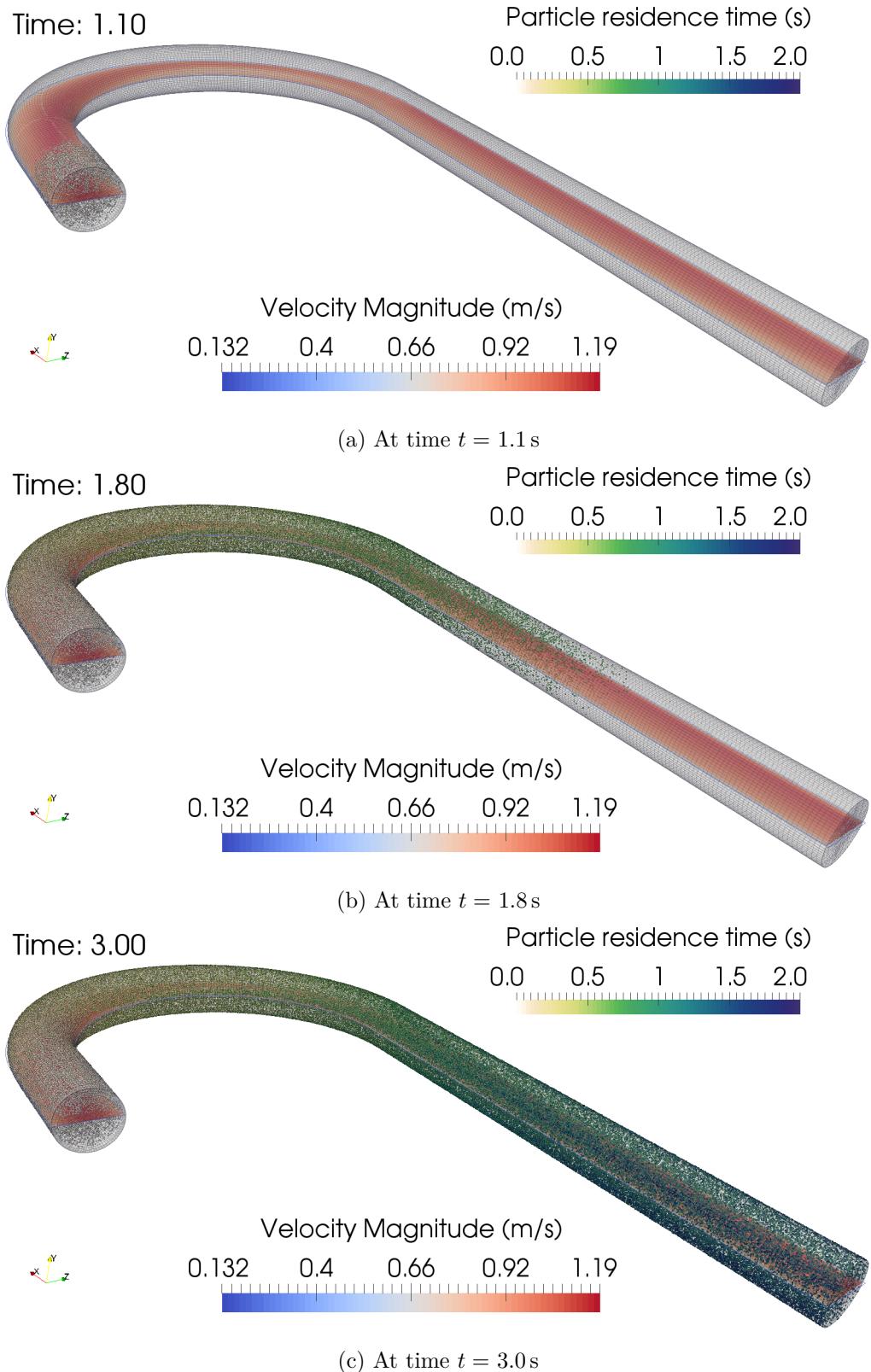


Figure 5.6: 3D view of the U-bend two-phase flow simulation showing both a slice of the fluid velocity inside the pipe and the particles (colored with the residence time) at various elapsed time.

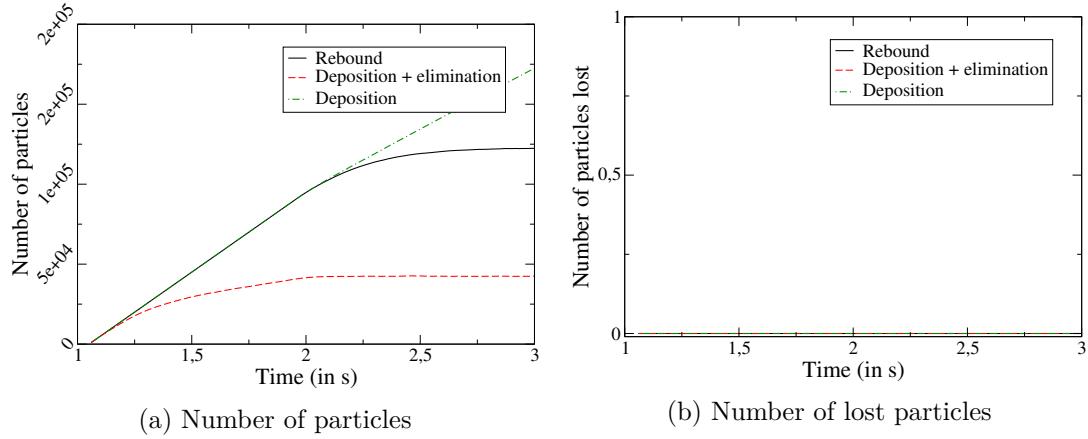


Figure 5.7: Evolution of the number of particles in the simulation (left) and the number of lost particles (right) for the three cases considered here for boundary conditions.

evolution of the number of particles: before  $t = 2$  s, it increases continuously as in the case with rebound, but it then continues to increase at a slightly slower rate due to the fact that particles are also exiting the domain.

Figure 5.7b displays the number of particles lost in the simulation: it further confirms the **precision and robustness** of the present algorithm since no particles are lost in the simulation even in such complex geometries.

## Conclusion

Overall, this algorithm for particle trajectography is precise and robust even in very complex geometries.

## 5.2 Adaptation of the multilayer deposition and resuspension models

### 5.2.1 Theory

The new model for multilayer particle deposition and resuspension has been detailed in the previous report and has been recently published in [1] (more details can be found there). It has been implemented in the new version of the Lagrangian module of *Code\_Saturne* (written in C).

### 5.2.2 Case studied

The case studied to test the multilayer deposition and resuspension model is the same as the one described in the previous report (more details can be found there).

### 5.2.3 Conclusion

This new implementation yields the same results for multilayer deposition and resuspension.

**Perspectives** There is a possibility to test the present model by performing simulations of large particle resuspension at various fluid velocities and comparing numerical results to the diagram of Shields. These diagrams provide information on the inception of rolling motion as well as on the near-wall motion of particles (whether they continuously roll on the surface or get away from the surface for a given period). These features should be captured by the present model (the difficulty might stem from the need to tune/play with the resuspension angle which has a direct consequence on saltation).

## 5.3 Corrections of the numerical schemes

### 5.3.1 Theory

**Particle dynamics** The dynamics of discrete particles and the corresponding system of SDEs has been described in details in [2] and only the main features are recalled here. The system of SDEs describing the dynamics of the discrete particles reads

$$\begin{cases} dx_{p,i}(t) = U_{p,i} dt, \\ dU_{p,i}(t) = \frac{1}{\tau_p} (U_{s,i} - U_{p,i}) dt + \mathcal{A}_i dt, \\ dU_{s,i}(t) = -\frac{1}{T_{L,i}^*} U_{s,i} dt + C_i dt + \sum_j B_{ij} dW_j(t), \end{cases} \quad (5.4)$$

where  $C_i$  is a term that includes all mean contributions: the mean pressure gradient,  $-(\partial \langle P \rangle / \partial x_i) / \rho_f$ , the mean drift term,  $(\langle U_{p,j} \rangle - \langle U_j \rangle)(\partial \langle U_i \rangle / \partial x_j)$ , and the mean part of the return-to-equilibrium term,  $\langle U_i \rangle / T_{L,i}^*$ .  $\mathcal{A}_i$  is an acceleration (gravity in the present work, but it can be extended for practical reasons to the case of other external force fields).

The weak numerical schemes, with the required features, are developed based on the analytical solution to Eqs. (5.4) *with constant coefficients* (independent of time), the main idea being to derive a numerical scheme by freezing the coefficients on the integration intervals. This methodology ensures *stability* and *consistency with all limit systems*:

- stability because the form of the equations gives analytical solutions with exponentials of the type  $\exp(-\Delta t/T)$  where  $T$  is one of the characteristic timescales ( $\tau_p$  and  $T_{L,i}^*$ ),
- consistency with all limit systems by construction, since the schemes are based on an analytical solution.

Different techniques shall be used to derive first and second-order (in time) schemes from the analytical solutions with constant coefficients. A first-order scheme can be obtained by computing, at each time step, the variables on the basis of the analytical solutions (all coefficients are frozen at the beginning of the integration interval), i.e. a numerical scheme of the *Euler* kind is obtained. A second-order scheme can be derived by resorting to a predictor-corrector technique where the prediction step is the first-order scheme.

**Analytical solution** Before presenting the weak numerical schemes, it is a prerequisite to give the analytical solutions to system (5.4), with constant coefficients (in time). These solutions are obtained by resorting to Itô's calculus in combination with the method of the variation of the constant. For instance, for the fluid velocity seen, one seeks a solution of the form  $U_{s,i}(t) = H_i(t) \exp(-t/T_i)$ , where  $H_i(t)$  is a stochastic process defined by (*from now on the notation is slightly changed:  $T_{L,i}^*$  is noted  $T_i$  for the sake of clarity in the complex formulae to come*)

$$dH_i(t) = \exp(t/T_i)[C_i dt + \check{B}_i dW_i(t)], \quad (5.5)$$

that is, by integration on a time interval  $[t_0, t]$  ( $\Delta t = t - t_0$ ),

$$\begin{aligned} U_{s,i}(t) &= U_{s,i}(t_0) \exp(-\Delta t/T_i) + C_i T_i [1 - \exp(-\Delta t/T_i)] \\ &\quad + \check{B}_i \exp(-t/T_i) \int_{t_0}^t \exp(s/T_i) dW_i(s), \end{aligned} \quad (5.6)$$

where  $\check{B}_i = B_{ii}$  since  $B_{ij}$  is a diagonal matrix). By proceeding in the same way for the other equations (position and velocity), the analytical solution is obtained for the entire system, cf. Table 5.3. The three stochastic integrals, Eqs. (5.10) to (5.12) in Table 5.3, are centred Gaussian processes. These integrals are defined implicitly, but they can be simplified by integration by parts, cf. Table 5.3.

**Weak first-order scheme** From the analytical solutions of this system assuming constant coefficients, the weak-first order scheme is extracted. The equations are described in Table 5.4.

**Weak second-order scheme** The weak second-order scheme consists in a correction step for the particle velocity and the velocity of the fluid seen as described in Table 5.5:

### 5.3.2 Verification case

To verify the present algorithm for the treatment of the equations of particle motion, numerical results are compared to analytical results in ideal cases that correspond to limiting cases (with constant coefficients).

Table 5.3: Analytical solutions to system (5.4) for time-independent coefficients.

$$\begin{aligned} x_{p,i}(t) &= x_{p,i}(t_0) + U_{p,i}(t_0)\tau_p[1 - \exp(-\Delta t/\tau_p)] + U_{s,i}(t_0)\theta_i\{T_i[1 - \exp(-\Delta t/T_i)] \\ &\quad + \tau_p[\exp(-\Delta t/\tau_p) - 1]\} + [C_i T_i]\{\Delta t - \tau_p[1 - \exp(-\Delta t/\tau_p)] \\ &\quad - \theta_i(T_i[1 - \exp(-\Delta t/T_i)] + \tau_p[\exp(-\Delta t/\tau_p) - 1])\} + \Omega_i(t) \end{aligned} \quad (5.7)$$

with  $\theta_i = T_i/(T_i - \tau_p)$

$$\begin{aligned} U_{p,i}(t) &= U_{p,i}(t_0)\exp(-\Delta t/\tau_p) + U_{s,i}(t_0)\theta_i[\exp(-\Delta t/T_i) - \exp(-\Delta t/\tau_p)] \\ &\quad + [C_i T_i]\{[1 - \exp(-\Delta t/\tau_p)] - \theta_i[\exp(-\Delta t/T_i) - \exp(-\Delta t/\tau_p)]\} \\ &\quad + \Gamma_i(t) \end{aligned} \quad (5.8)$$

$$U_{s,i}(t) = U_{s,i}(t_0)\exp(-\Delta t/T_i) + C_i T_i[1 - \exp(-\Delta t/T_i)] + \gamma_i(t) \quad (5.9)$$

The stochastic integrals  $\gamma_i(t)$ ,  $\Gamma_i(t)$ ,  $\Omega_i(t)$  are given by:

$$\gamma_i(t) = \check{B}_i \exp(-t/T_i) \int_{t_0}^t \exp(s/T_i) dW_i(s), \quad (5.10)$$

$$\Gamma_i(t) = \frac{1}{\tau_p} \exp(-t/\tau_p) \int_{t_0}^t \exp(s/\tau_p) \gamma_i(s) ds, \quad (5.11)$$

$$\Omega_i(t) = \int_{t_0}^t \Gamma_i(s) ds. \quad (5.12)$$

By resorting to stochastic integration by parts,  $\gamma_i(t)$ ,  $\Gamma_i(t)$ ,  $\Omega_i(t)$  can be written:

$$\gamma_i(t) = \check{B}_i \exp(-t/T_i) I_{1,i}, \quad (5.13)$$

$$\Gamma_i(t) = \theta_i \check{B}_i [\exp(-t/T_i) I_{1,i} - \exp(-t/\tau_p) I_{2,i}], \quad (5.14)$$

$$\begin{aligned} \Omega_i(t) &= \theta_i \check{B}_i \{(T_i - \tau_p) I_{3,i} \\ &\quad - [T_i \exp(-t/T_i) I_{1,i} - \tau_p \exp(-t/\tau_p) I_{2,i}]\}, \end{aligned} \quad (5.15)$$

$$\text{with } I_{1,i} = \int_{t_0}^t \exp(s/T_i) dW_i(s), \quad I_{2,i} = \int_{t_0}^t \exp(s/\tau_p) dW_i(s)$$

$$\text{and } I_{3,i} = \int_{t_0}^t dW_i(s).$$

Table 5.4: Weak first-order scheme (Euler scheme)

Numerical integration of the system:

$$\begin{aligned}x_{p,i}^{n+1} &= x_{p,i}^n + A_1 U_{p,i}^n + B_1 U_{s,i}^n + C_1 [T_i^n C_i^n] + \Gamma_i^n, \\U_{p,i}^{n+1} &= U_{p,i}^n \exp(-\Delta t/\tau_p^n) + D_1 U_{s,i}^n + [T_i^n C_i^n](E_1 - D_1) + \Omega_i^n, \\U_{s,i}^{n+1} &= U_{s,i}^n \exp(-\Delta t/T_i^n) + [T_i^n C_i^n][1 - \exp(-\Delta t/T_i^n)] + \gamma_i^n.\end{aligned}$$

The coefficients  $A_1$ ,  $B_1$ ,  $C_1$ ,  $D_1$  and  $E_1$  are given by:

$$\begin{aligned}A_1 &= \tau_p^n [1 - \exp(-\Delta t/\tau_p^n)], \\B_1 &= \theta_i^n [T_i^n (1 - \exp(-\Delta t/T_i^n)) - A_1] \quad \text{with} \quad \theta_i^n = T_i^n / (T_i^n - \tau_p^n), \\C_1 &= \Delta t - A_1 - B_1, \\D_1 &= \theta_i^n [\exp(-\Delta t/T_i^n) - \exp(-\Delta t/\tau_p^n)], \\E_1 &= 1 - \exp(-\Delta t/\tau_p^n).\end{aligned}$$

The stochastic integrals  $\gamma_i^n$ ,  $\Omega_i^n$ ,  $\Gamma_i^n$  are simulated by:

$$\begin{aligned}\gamma_i^n &= P_{11} \mathcal{G}_{1,i}, \\ \Omega_i^n &= P_{21} \mathcal{G}_{1,i} + P_{22} \mathcal{G}_{2,i}, \\ \Gamma_i^n &= P_{31} \mathcal{G}_{1,i} + P_{32} \mathcal{G}_{2,i} + P_{33} \mathcal{G}_{3,i},\end{aligned}$$

where  $\mathcal{G}_{1,i}$ ,  $\mathcal{G}_{2,i}$ ,  $\mathcal{G}_{3,i}$  are independent  $\mathcal{N}(0, 1)$  random variables.

The coefficients  $P_{11}$ ,  $P_{21}$ ,  $P_{22}$ ,  $P_{31}$ ,  $P_{32}$ ,  $P_{33}$  are defined as:

$$\begin{aligned}P_{11} &= \sqrt{\langle (\gamma_i^n)^2 \rangle}, \\P_{21} &= \frac{\langle \Gamma_i^n \gamma_i^n \rangle}{\sqrt{\langle (\gamma_i^n)^2 \rangle}}, \quad P_{22} = \sqrt{\langle (\Gamma_i^n)^2 \rangle - \frac{\langle \Gamma_i^n \gamma_i^n \rangle^2}{\langle (\gamma_i^n)^2 \rangle}}, \\P_{31} &= \frac{\langle \Omega_i^n \gamma_i^n \rangle}{\sqrt{\langle (\gamma_i^n)^2 \rangle}}, \quad P_{32} = \frac{1}{P_{22}} (\langle \Omega_i^n \Gamma_i^n \rangle - P_{21} P_{31}), \quad P_{33} = \sqrt{\langle (\Omega_i^n)^2 \rangle - P_{31}^2 - P_{32}^2}.\end{aligned}$$

Table 5.5: Weak second-order scheme

---

Prediction step: Euler scheme, see Table 5.4 (predicted values noted with  $\sim$ ).

Correction step:

$$\begin{aligned} U_{p,i}^{n+1} &= \frac{1}{2} U_{p,i}^n \exp(-\Delta t/\tau_p^n) + \frac{1}{2} U_{p,i}^n \exp(-\Delta t/\tilde{\tau}_p^{n+1}) \\ &\quad + \frac{1}{2} U_{s,i}^n C_{2c}(\tau_p^n, T_i^n) + \frac{1}{2} U_{s,i}^n C_{2c}(\tilde{\tau}_p^{n+1}, \tilde{T}_i^{n+1}) \\ &\quad + A_{2c}(\tau_p^n, T_i^n) [T_i^n C_i^n] + B_{2c}(\tilde{\tau}_p^{n+1}, \tilde{T}_i^{n+1}) [\tilde{T}_i^{n+1} C_i^{n+1}] \\ &\quad + A_2(\Delta t, \tau_p^n) [\tau_p^n A_i^n] + B_2(\Delta t, \tilde{\tau}_p^{n+1}) [\tilde{\tau}_p^{n+1} A_i^{n+1}] + \tilde{\Gamma}_i^{n+1}, \\ U_{s,i}^{n+1} &= \frac{1}{2} U_{s,i}^n \exp(-\Delta t/T_i^n) + \frac{1}{2} U_{s,i}^n \exp(-\Delta t/\tilde{T}_i^{n+1}) + A_2(\Delta t, T_i^n) [T_i^n C_i^n] \\ &\quad + B_2(\Delta t, \tilde{T}_i^{n+1}) [\tilde{T}_i^{n+1} C_i^{n+1}] + \tilde{\gamma}_i^{n+1}. \end{aligned}$$

The coefficients  $A_2$ ,  $B_2$ ,  $A_{2c}$ ,  $B_{2c}$  et  $C_{2c}$  are defined as:

$$A_2(\Delta t, x) = -\exp(-\Delta t/x) + [1 - \exp(-\Delta t/x)][x/\Delta t],$$

$$B_2(\Delta t, x) = 1 - [1 - \exp(-\Delta t/x)][x/\Delta t],$$

$$A_{2c}(x, y) = -\exp(-\Delta t/x) + [(x+y)/\Delta t][1 - \exp(-\Delta t/x)] - (1+y/\Delta t) C_{2c}(x, y),$$

$$B_{2c}(x, y) = 1 - [(x+y)/\Delta t][1 - \exp(-\Delta t/x)] + (y/\Delta t) C_{2c}(x, y),$$

$$C_{2c}(x, y) = [y/(y-x)][\exp(-\Delta t/y) - \exp(-\Delta t/x)].$$

The stochastic integrals  $\tilde{\gamma}_i^{n+1}$  and  $\tilde{\Gamma}_i^{n+1}$  are simulated as follows:

$$\begin{aligned} \tilde{\gamma}_i^{n+1} &= \sqrt{\frac{[B_i^*]^2 \tilde{T}_i^{n+1}}{2} [1 - \exp(-2\Delta t/\tilde{T}_i^{n+1})]} \mathcal{G}_{1,i}, \\ \text{with } & \left[ 1 - \exp(-2\Delta t/\tilde{T}_i^{n+1}) \right] B_i^* = A_2(2\Delta t, \tilde{T}_i^{n+1}) \sqrt{(\check{B}_i^n)^2} + \\ & B_2(2\Delta t, \tilde{T}_i^{n+1}) \sqrt{(\check{\tilde{B}}_i^{n+1})^2}. \end{aligned}$$

$$\tilde{\Gamma}_i^{n+1} = \frac{\langle \tilde{\Gamma}_i^{n+1} \tilde{\gamma}_i^{n+1} \rangle}{\langle (\tilde{\gamma}_i^{n+1})^2 \rangle} \tilde{\gamma}_i^{n+1} + \sqrt{\langle (\tilde{\Gamma}_i^{n+1})^2 \rangle - \frac{\langle \tilde{\Gamma}_i^{n+1} \tilde{\gamma}_i^{n+1} \rangle^2}{\langle (\tilde{\gamma}_i^{n+1})^2 \rangle}} \mathcal{G}_{2,i}$$

$$\text{with } \langle \tilde{\Gamma}_i^{n+1} \tilde{\gamma}_i^{n+1} \rangle = \langle \Gamma_i \gamma_i \rangle (\tau_p^n, \tilde{T}_i^{n+1}, B_i^*) \quad \text{and} \quad \langle (\tilde{\Gamma}_i^{n+1})^2 \rangle = \langle \Gamma_i^2 \rangle (\tau_p^n, \tilde{T}_i^{n+1}, B_i^*).$$


---

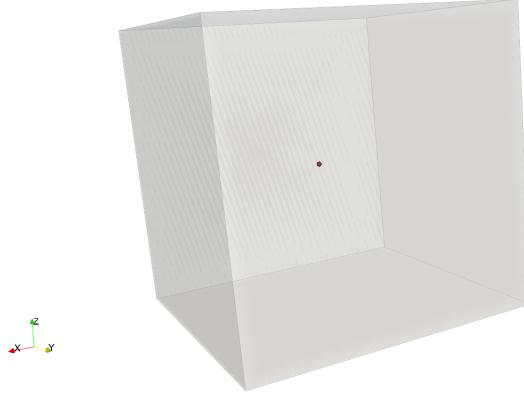


Figure 5.8: 3D view of the single cell used for the verification of the numerical scheme in ideal cases.

### Case description

This test case consists in an isotropic case as the one described in [3]: constant coefficients  $C_i(t, \mathbf{x}_p) = 0$ ,  $\mathbf{A}_i(t, \mathbf{x}_p) = 0$  and initial conditions  $x_p(0) = U_p(0) = U_s(0) = 0$ . In that case, the system simplifies to

$$\begin{cases} dx_p = U_p dt \\ dU_p = \frac{1}{\tau_p} (U_s - U_p) dt \\ dU_s = -\frac{1}{T} U_s dt + \sigma dW(t), \end{cases} \quad (5.16)$$

and the solution of the system is

$$\begin{cases} x_p = \Omega(t) \\ U_p = \Gamma(t) \\ U_s = \gamma(t), \end{cases} \quad (5.17)$$

### Computational domain

To perform a numerical simulation in such an ideal case, a very simple computational domain consisting of a single cubic cell (with a size of 10 m) has been retained.

### Physical model for the fluid flow

**Generalities** Since all parameters are fixed manually in the Lagrangian module of *Code\_Saturne*, the simulation of the fluid phase is not primordial. Nevertheless, the following parameters have been retained to be able to run a two-phase flow simulation. The flow is isothermal and incompressible. Gravity has been neglected. The properties used for the liquid in the flow domain are summarised in Table 5.6:

$U_{inlet}$	$\mu_f$	$\rho_f$
$0 \text{ m s}^{-1}$	$1.002 \times 10^{-3} \text{ kg m}^{-1} \text{s}^{-1}$	$998 \text{ kg m}^{-3}$

Table 5.6: Fluid properties used in the simulation: inlet velocity  $U_{inlet}$ , Reynolds number  $Re$ , dynamic viscosity  $\mu_f$  and density  $\rho_f$ .

**Turbulence models** A turbulence model has been activated to be able to use the particle turbulent dispersion model:

- Turbulence model:  $k - \varepsilon$  (with default values)
- Unsteady flow

### Initial conditions

The initial conditions are the following:

- reference and initial pressure: 101 325 Pa
- fluid velocity:  $0.0 \text{ m s}^{-1}$
- temperature: 293.15 K

### Boundary conditions

**Inlet boundary** An inlet boundary condition is needed for the particle injection. As a result, one of the faces of the cube is used as an inlet and the fluid is injected with the following properties:

- velocity:  $0.0 \text{ m s}^{-1}$
- hydraulic diameter (for turbulence initialisation): 1.0 m

**Wall boundary** The other boundary conditions are fixed to symmetry.

### Physical model for particles

**Injection** Particles are injected in the domain at the inlet. The properties used for the injected particles are:

- Particle diameter:  $1 \times 10^{-3} \text{ m}$
- Monodispersed particles

- Particle density:  $998 \text{ kg m}^{-3}$
- Frequency of injection: 0 (only initially)
- Number of particles in class: 20 000

**Boundary condition** The wall boundary conditions for particles have been set to

- symmetry (particles zero-flux)

**Model for transport** The CFD simulation with the injection of particles has been performed using the Lagrangian module in *Code\_Saturne* and the following properties

- Eulerian-Lagrangian multi-phase treatment: one way coupling
- The continuous phase is a steady flow
- No additional models associated with the particles
- Integration for the stochastic differential equations: first-order scheme or second-order scheme
- Particle turbulent dispersion model: activated

**Numerical parameters** The two-phase flow simulation has been performed using the following numerical parameters:

- Calculation restart from the fluid flow simulation
- Constant time step: 0.001 s
- Number of iterations: 2000 or 4000

The main subroutine of the Lagrangian module (`cs_lagr.c`) has been modified to enforce fixed coefficients ( $C_i(t, \mathbf{x}_p) = 0$ ,  $\mathbf{A}_i(t, \mathbf{x}_p) = 0$ ,  $T_L$ ,  $\tau_p$ ) throughout the calculation while the injection (in `cs_lagr_new.c`) has also been modified to have all the particles starting from the middle of the simulation (see also Figure 5.8). The various cases studied are summarised in Table 5.7:

## Results

Two-phase flow simulations of system 5.16 have been performed in the various cases previously mentioned. The system considered consists in purely diffusive motion of particles, as seen in Figure 5.9. Results are displayed in Figures 5.10-5.13 for the second-order moments (first-order moments are omitted since the solutions of system 5.16 are Gaussian random variables with a mean equal to zero) in the various cases. It can be seen that all numerical results are close to the analytical solution.

Case	$\tau_p$ (s)	$T_L$ (s)	$\sigma$ ( $\text{m s}^{-3/2}$ )
General case	$10^{-1}$	$2 \times 10^{-1}$	$10^1$
Limit case I	$10^{-5}$	$10^{-1}$	$10^1$
Limit case II	$10^{-1}$	$10^{-5}$	$10^3$
Limit case III	$2 \times 10^{-5}$	$10^{-5}$	$10^3$

Table 5.7: Values of the fixed parameters in the various cases studied. General case:  $\Delta t \ll T_L, \tau_p$ . Limit case I:  $\tau_p \ll \Delta t \ll T_L$ . Limit case II:  $T_L \ll \Delta t \ll \tau_p$ . Limit case III:  $\tau_p, T_L \ll \Delta t$ .



Figure 5.9: 3D view of the particles inside single cell (zoom) at the initial and final time step.

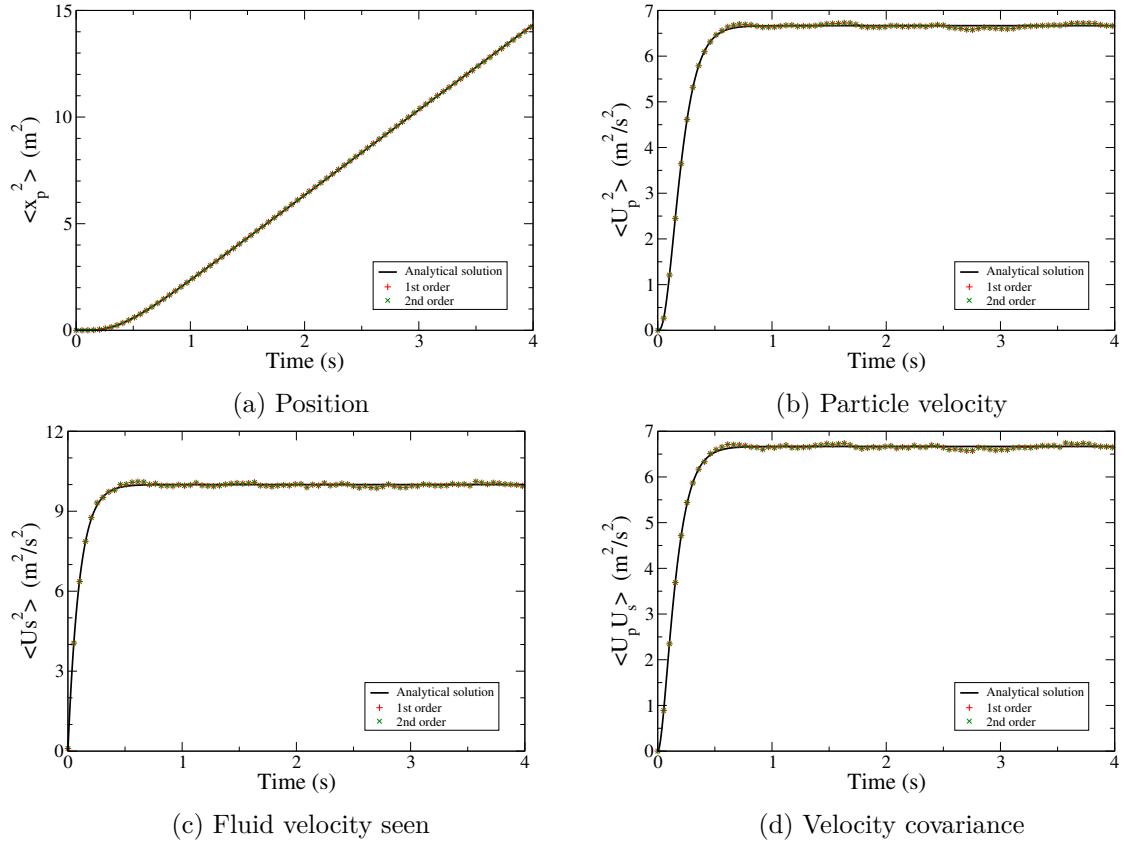


Figure 5.10: Comparison of numerical results for system 5.16 (symbols for 1<sup>st</sup> and 2<sup>nd</sup> order schemes) and analytical solutions (lines) in the general case.

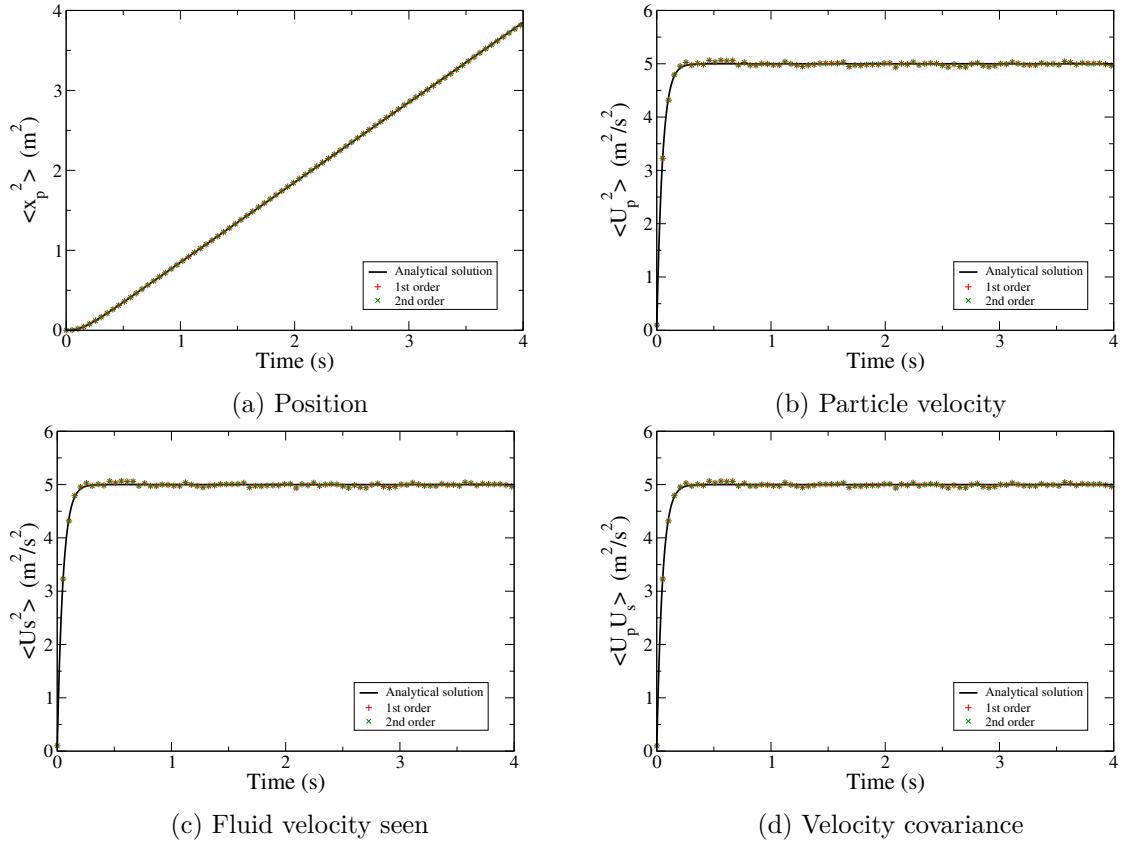


Figure 5.11: Comparison of numerical results for system 5.16 (symbols for 1<sup>st</sup> and 2<sup>nd</sup> order schemes) and analytical solutions (lines) in the limit case I.

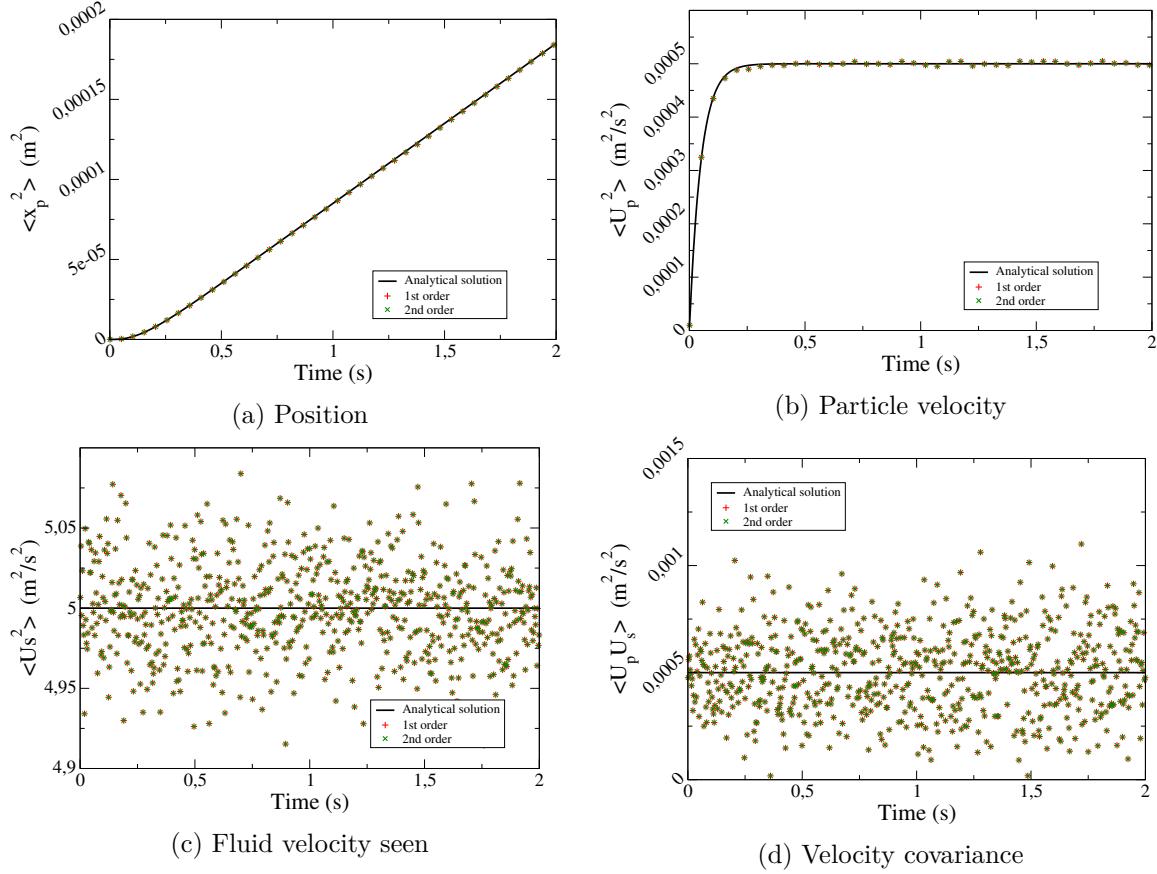


Figure 5.12: Comparison of numerical results for system 5.16 (symbols for 1<sup>st</sup> and 2<sup>nd</sup> order schemes) and analytical solutions (lines) in the limit case II.

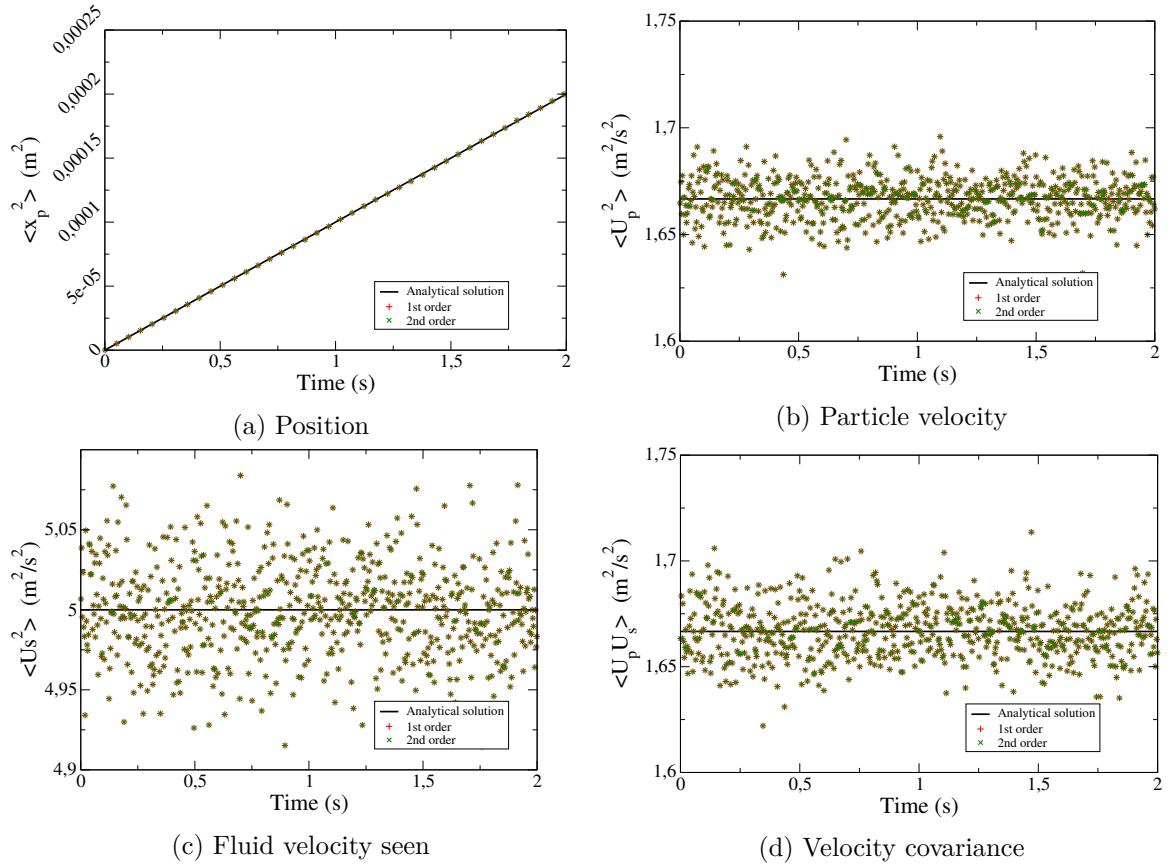


Figure 5.13: Comparison of numerical results for system 5.16 (symbols for 1<sup>st</sup> and 2<sup>nd</sup> order schemes) and analytical solutions (lines) in the limit case III.

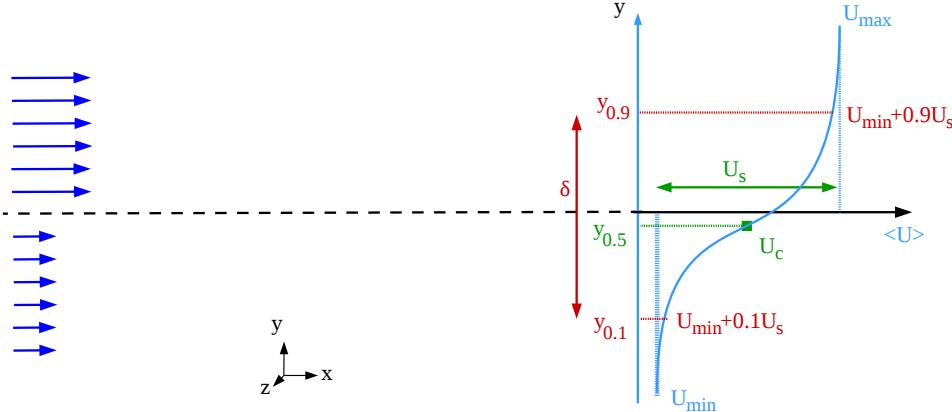


Figure 5.14: Sketch of a plane 2D mixing layer showing the characteristic convective velocity  $U_c$  and the characteristic velocity difference  $U_s$ , the characteristic width of the flow  $\delta(x)$  and reference lateral position  $\bar{y}(x)$ .

## Conclusion

Overall, this algorithm for the numerical scheme is precise and robust even in all the ideal cases considered.

### 5.3.3 Validation case

To validate the present algorithm for particle equations of motion, a simulation has been performed in the case of a mixing layer loaded with fluid particles.

## Experimental data

The case of mixing layers corresponds to the turbulent flow formed between two uniform, nearly parallel streams of different velocities  $U_{max}$  and  $U_{min}$ . Figure 5.14 displays the mixing layer created by the interaction between two streams with different velocities. In that case, the dominant direction of flow is  $x$ , the cross-stream coordinate is  $y$  and statistics are independent of the spanwise coordinate,  $z$ . The mean velocity can thus be written  $\langle U(x, y, z) \rangle = \langle U(x, y) \rangle$ . Numerous studies have shown that the flow is self-similar and depends on characteristic parameters (more details can be found in [4, 5]):

- the characteristic convection velocity

$$U_c = \frac{1}{2} (U_{max} + U_{min}) \quad (5.18)$$

- the characteristic velocity difference

$$U_s = (U_{max} - U_{min}) \quad (5.19)$$

- the characteristic width of the flow

$$\delta(x) = y_{0.9}(x) - y_{0.1}(x) \quad (5.20)$$

defined here as the extent of the region where the fluid velocity ranges from  $U_{min} + 0.1U_s$  to  $U_{min} + 0.9U_s$  (see Figure 5.14).

- the characteristic reference lateral position

$$\bar{y}(x) = \frac{1}{2} (y_{0.9}(x) + y_{0.1}(x)) \quad (5.21)$$

These characteristic values allow to define a scaled velocity

$$f(\xi) = \frac{\langle U \rangle - U_c}{U_s} \quad (5.22)$$

that depends only on the scaled cross-stream location (self-similar flow)

$$\xi = \frac{y - \bar{y}(x)}{\delta(x)} \quad (5.23)$$

Similarly, the Reynolds stresses have been shown to be self-similar when scaled by  $U_s^2$ . Experimental data have also shown that the spreading parameter  $S$  defined as

$$S = \frac{U_c}{U_s} \frac{d\delta}{dx} \quad (5.24)$$

is independent of the velocity ratio and thus constant. The range of reported values goes from  $S \approx 0.06$  to  $S = 0.11$ .

## Case description

In the following, numerical results of a mixing layer are compared to available experimental data. For that purpose, the test case considered consists in a flow within a 2D rectangular duct configuration with two different inlet velocities and loaded with fluid particles.

## Computational domain

The 2D CFD geometry of the computational domain is shown in Figure (5.15). It consists of a rectangular duct with the following properties:

- Length: 10 m (100 cells)
- Height: 0.8 m (100 cells)

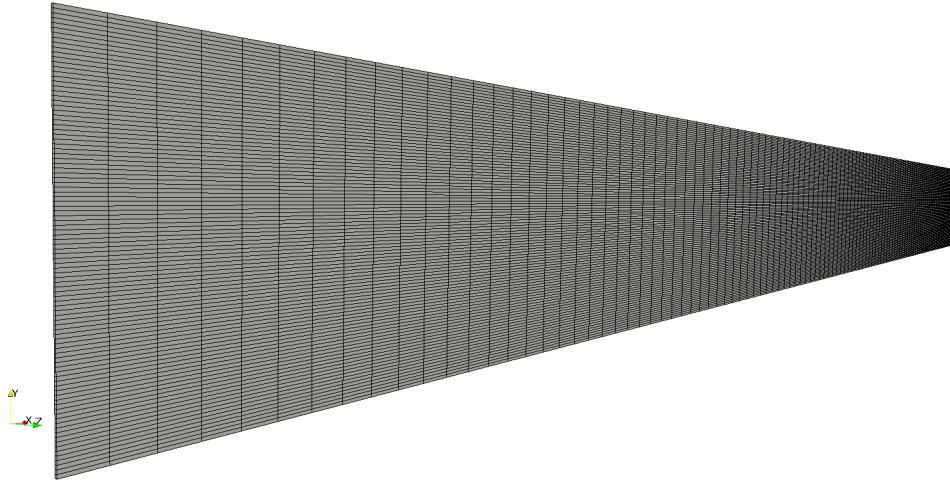


Figure 5.15: 2D view of the geometry of the computational domain for the mixing layer simulation.

$U_{inlet}$	$\mu_f$	$\rho_f$
$1 \text{ m s}^{-1}$	$1.002 \times 10^{-3} \text{ kg m}^{-1} \text{s}^{-1}$	$998 \text{ kg m}^{-3}$

Table 5.8: Fluid properties used in the simulation: inlet velocity  $U_{inlet}$ , dynamic viscosity  $\mu_f$  and density  $\rho_f$ .

- Width: 0.004 m (1 cell)

The grid comprises 10 000 rectangular uniform cells.

### Physical model for the fluid flow

**Generalities** The flow is isothermal and incompressible. Gravity has been neglected. The properties used for the liquid in the flow domain are summarised in Table 5.8:

**Turbulence models** A CFD simulation of the turbulent flow in this U-bend 3D configuration has been performed using *Code\_Saturne*. For that purpose, we have retained:

- Turbulence model:  $Rij - \varepsilon$  LRR,  $Rij - \varepsilon$  SSG or  $Rij - \varepsilon$  Rotta (with default values)
- Unsteady flow

### Initial conditions

The initial conditions are the following:

- reference and initial pressure: 101 325 Pa

- fluid velocity:  $0.0 \text{ m s}^{-1}$
- temperature:  $300 \text{ K}$

## Boundary conditions

**Inlet boundary** At the inlet, the flow is divided in two regions which have a uniform profile with the following properties:

- velocity:  
 $2.0 \text{ m s}^{-1}$  for  $y < 0.4 \text{ m}$   
 $3.0 \text{ m s}^{-1}$  for  $y > 0.4 \text{ m}$
- turbulent intensity (for turbulence initialisation):  
intensity 1 % and hydraulic diameter 0.4 m for  $y < 0.4 \text{ m}$   
intensity 0.44 % and hydraulic diameter 0.4 m for  $y > 0.4 \text{ m}$

**Wall boundary** The wall boundary condition is a symmetry.

**Outlet boundary** No imposed quantity at the outlet.

## Numerical parameters

The fluid-flow simulation has been performed using the following numerical parameters:

- Constant time step: 0.01 s
- Number of iterations: 4000

These parameters ensure the following properties

Field	Minimum	Maximum	Mean
Courant number	0.20	0.30	0.25
Fourier number	0.013	2240	1.83

Table 5.9: Properties resulting from the parameters used in the simulation (Cournat, Fourier numbers).

## Physical model for particles

**Injection** Fluid particles have been injected in the domain at the inlet. The properties used for the injected particles are:

- Particle diameter:  $1 \times 10^{-6}$  m
- Monodispersed particles
- Particle density:  $998 \text{ kg m}^{-3}$
- Frequency of injection: 1 (every time step)
- Number of particles in class:
  - 80 for  $y < 0.4 \text{ m}$
  - 120 for  $y > 0.4 \text{ m}$

**Boundary condition** The wall boundary conditions for particles have been set to

- Symmetry (zero-flux)

**Model for transport** The CFD simulation with the injection of particles has been performed using the Lagrangian module in *Code\_Saturne* and the following properties

- Eulerian-Lagrangian multi-phase treatment: one way coupling
- No additional models associated with the particles
- Turbulent deposition: particle deposition sub-model deactivated
- Integration for the stochastic differential equations: first-order scheme
- Particle turbulent dispersion model: activated
- Suppresses the crossing trajectory effect: activated
- Complete model for turbulent dispersion: activated (starting from iteration 4500 in  $x$  direction).

**Numerical parameters** The two-phase flow simulation has been performed using the following numerical parameters:

- Calculation restart from the fluid flow simulation (frozen velocity and pressure fields)
- Constant time step: 0.01 s
- Number of iterations: 8000

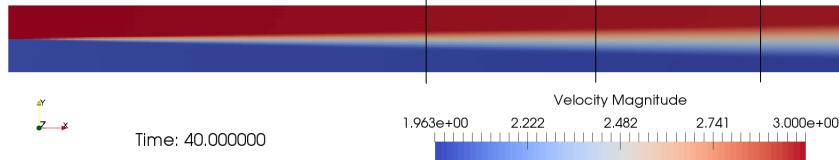


Figure 5.16: Snapshot of the fluid velocity obtained from the fluid-flow simulation inside the 2D rectangular duct showing the formation and growth of the mixing layer (vertical lines are used to extract and plot data on specific regions).

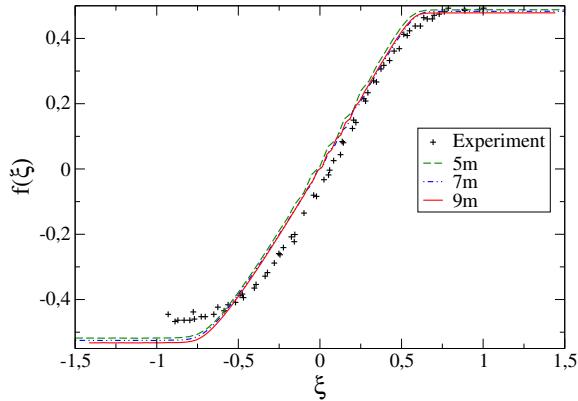


Figure 5.17: Scaled mean velocity profile in a plane mixing layer: comparison between experimental data (symbols) and numerical results sampled at various distances from the inlet (lines).

- Statistics:
  - statistics start from iteration 4000
  - Start of steady-state statistics from iteration 4500
  - Threshold for statistical weight 0.0

The user subroutine `cs_user_lagr_model.c` has also been modified to activate the calculation of statistics related to the velocity seen by the particles.

## Results

**Single-phase flow simulation** First, we compare the numerical results obtained with the single-phase flow simulation to the experimental data (more details in [4, 5]). The velocity field inside the rectangular duct is displayed in Figure 5.16: it can be seen that the mixing leads to the growth of a turbulent region with intermediate velocities.

To further characterise the fluid flow, we have extracted and plotted data along the vertical lines located at various  $x$  positions: 5 m, 7 m and 9 m from the inlet (see also Figure 5.16). Figure 5.17 displays the scaled mean velocity profile. It can be seen that numerical results are in relatively good agreement with experimental data and that self-similarity is also captured by numerical simulations. This is further confirmed by the scaled Reynolds stresses, which are

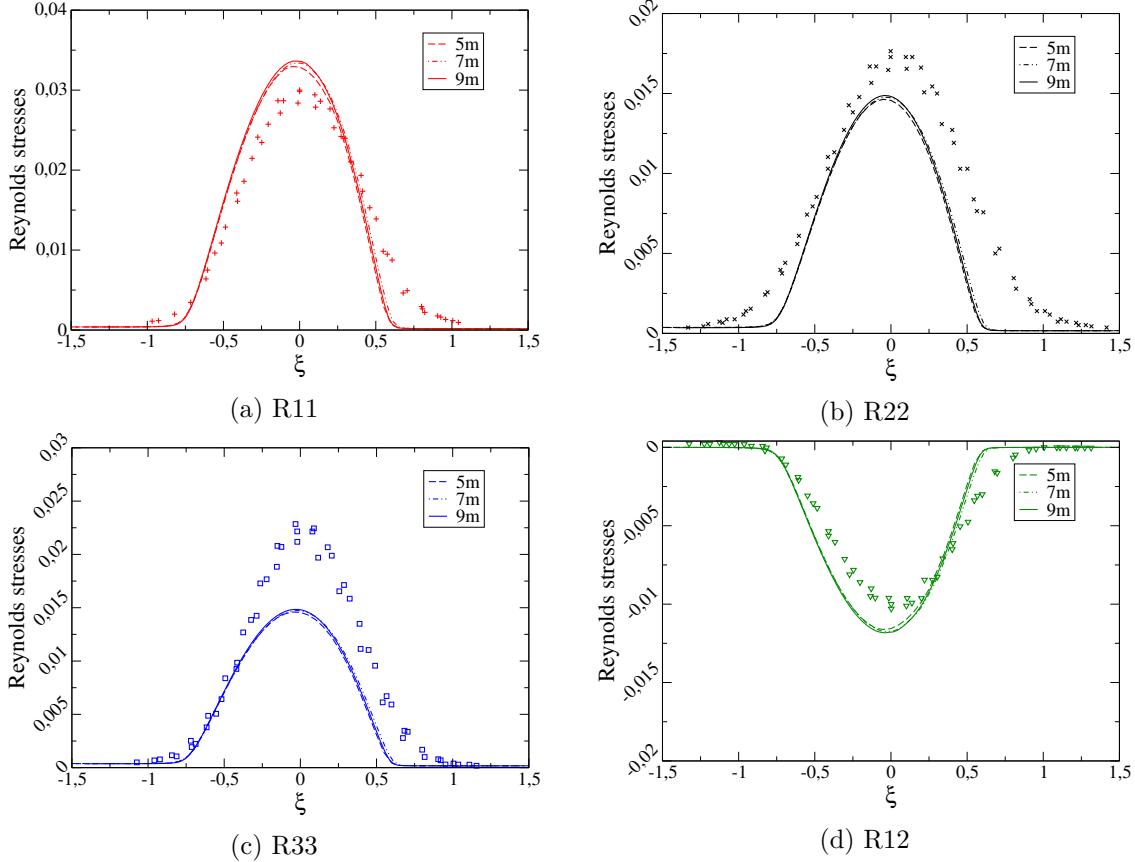
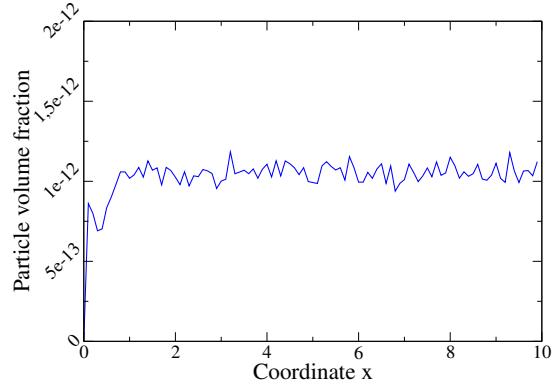


Figure 5.18: Scaled Reynolds stresses in a plane mixing layer: comparison between experimental data (symbols) and numerical results sampled at various distances from the inlet (lines).

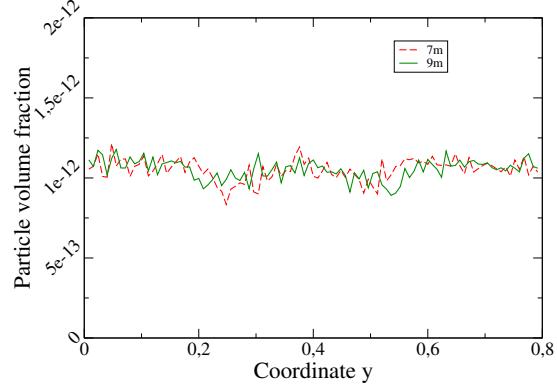
plotted in Figure 5.18 (R33 is not very satisfactory but it should be remembered that the present calculations are 2D whereas experiments were done in a 3D case). The value of the spreading parameter  $S = 0.068$  is also within the range of values measured experimentally.

**Two-phase flow simulations** Second, fluid particles have been injected in the simulation to check the robustness of the present algorithm for the treatment of the equations of particle motion. For that purpose, numerical results for the particle-phase are compared to numerical results for the fluid-phase but a special attention has been devoted to check that the volume fraction of particles remains homogeneous in the simulation (fluid particles should not accumulate in specific regions).

A first simulation has been performed using the algorithm described previously. It can be seen in Figure 5.19 that the volume fraction reaches quickly a constant value in the middle of the rectangular duct ( $x = 0.4$  m) but that it fluctuates in the  $y$  direction (with lower values in regions corresponding to the laminar-turbulent transition). This is confirmed by a visual analysis of the volume fraction within the computational domain (see Figure 5.20). Besides, the mean particle



(a) Horizontal section ( $x = 0.4$  m).



(b) Vertical sections)

Figure 5.19: Particle volume fraction in the plane mixing layer sampled along the middle of the duct and at various distances from the inlet.

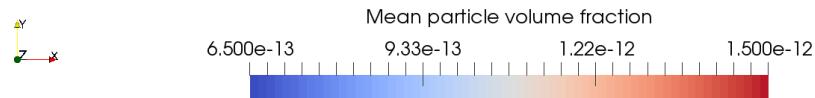


Figure 5.20: Particle volume fraction in the plane mixing layer showing the depletion region.

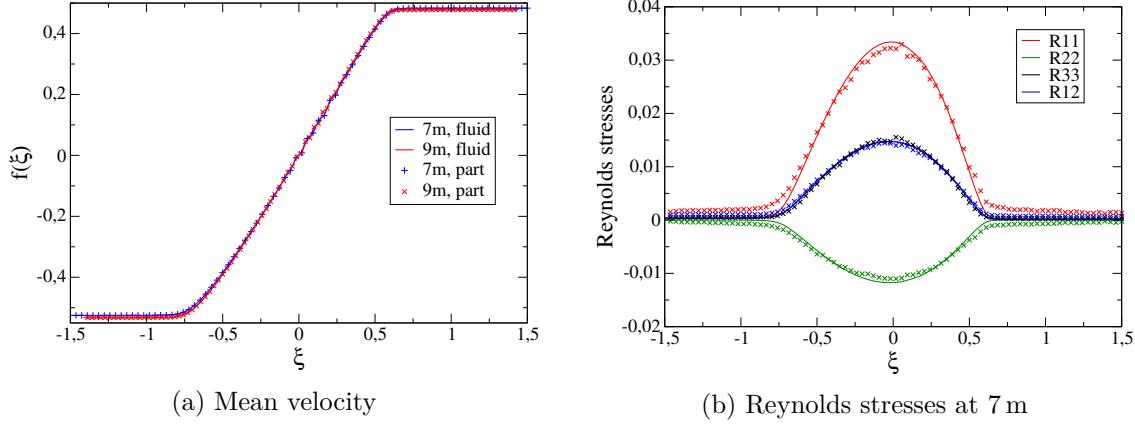


Figure 5.21: Scaled mean velocity and Reynolds stresses profiles in a plane mixing layer: comparison between numerical results for the fluid phase (lines) and for the dispersed phase (symbols) sampled at various distances from the inlet.

velocities and Reynolds stresses have been compared to the numerical results for the fluid phase in Figure 5.21a and 5.21b: it can be seen that the mean velocity of fluid particles is comparable but that the Reynolds stresses of the fluid particles in the laminar regions is much higher than expected (significant non-zero values).

Further simulations have been performed to investigate the origin of these discrepancies (i.e. non-homogeneous fluid particle volume fraction and high Reynolds stresses in the laminar region). Similarly to what is usually done in the simulations of the fluid phase, a correction has been added to account for the incompressibility condition at the end of the Lagrangian time step. Here, it has been simplified by an instantaneous return-to-equilibrium term to ensure that the mean velocity of the fluid seen by the particles inside each cell is equal to the mean velocity of the fluid (which satisfies the incompressibility condition). We thus enforced for each particle labelled  $i$  inside a cell:

$$U_{s,i} = U_{s,i} - (\langle U_s \rangle - \langle U_f \rangle) \quad (5.25)$$

Numerical results are shown in Figures 5.24 and 5.22 where it can be seen that the volume fraction is now homogeneous over the whole simulation domain (except close to the inlet possibly due to injection conditions, see also Figure 5.23). Therefore, it appears that this correction ensures an homogeneous repartition of fluid particles within the domain. Meanwhile, the mean velocity and Reynolds stresses are barely affected by these modifications (not shown here).

The presence of non-negligible Reynolds stresses in the laminar regions has to do with the ratio  $k/\varepsilon$  given by the calculation of the fluid phase. Indeed, the fluid Lagrangian integral timescale  $T_L$  (directly proportional to  $k/\varepsilon$ ) can reach relatively high values in the laminar regions as depicted in Figure 5.25. As a result, the particles exiting the turbulent mixing layer can retain a high energy level for a significantly long time, leading to the deviations seen in Figure 5.21b. A numerical simulation has been performed enforcing  $T_L = 0.01$  when  $R_{11} < 0.001$

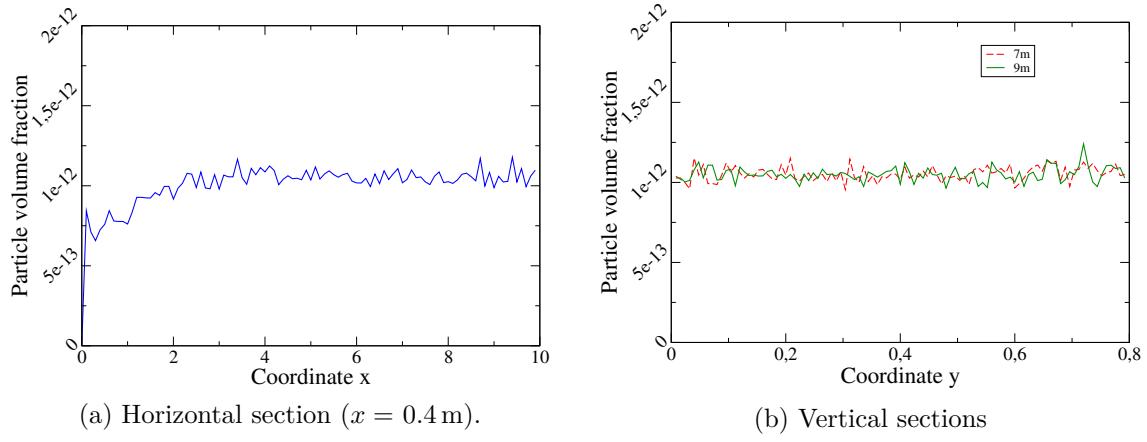


Figure 5.22: Particle volume fraction in the plane mixing layer sampled along the middle of the duct and at various distances from the inlet with additional corrections for the incompressibility condition.

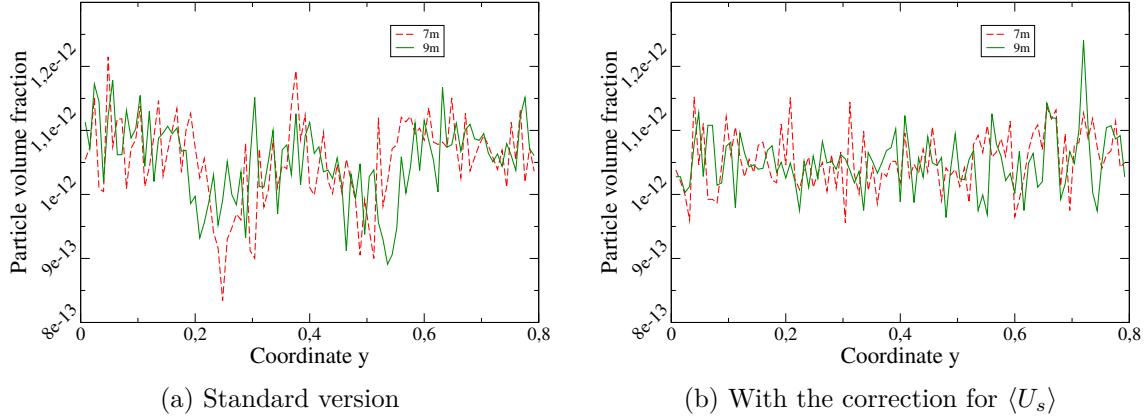


Figure 5.23: Particle volume fraction in the plane mixing layer at various distances from the inlet: comparison between the standard version and the one with the correction for  $\langle U_s \rangle$ .

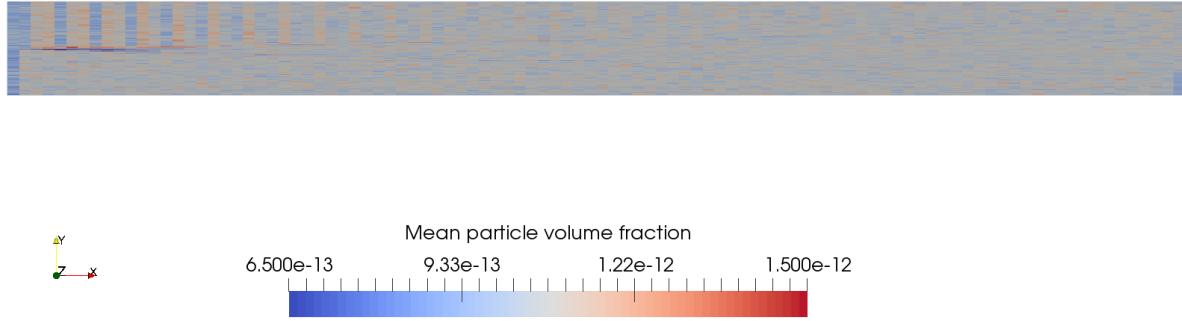


Figure 5.24: Particle volume fraction in the plane mixing layer with additional corrections for the incompressibility condition.

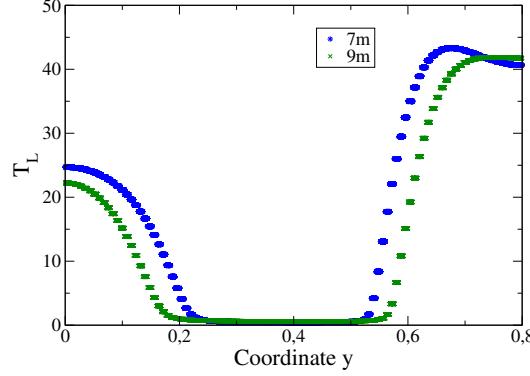


Figure 5.25: Value of the fluid Lagrangian integral timescale  $T_L$  in the plane mixing layer at various distances from the inlet.

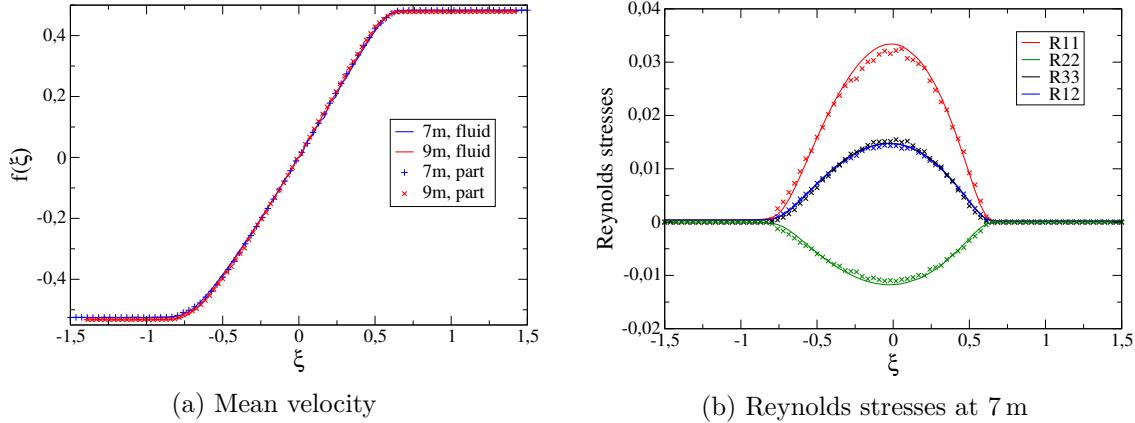


Figure 5.26: Scaled mean velocity and Reynolds stresses profiles in a plane mixing layer with a correction for  $T_L$ : comparison between numerical results for the fluid phase (lines) and for the dispersed phase (symbols) sampled at various distances from the inlet.

(this corresponds more or less to the laminar-turbulent transition). In that case, the Reynolds stresses decrease as expected inside the laminar region and numerical results for the particle phase are in good agreement with results for the fluid phase (see Figure 5.26). However, as seen in Figure 5.26, fluctuations in the volume fraction across the laminar regions are visible (probably due to injection conditions).

A simulation with both the correction for the incompressibility condition and the correction for  $T_L$  provides similar results as the one obtained with the correction for  $T_L$  only (not shown here).

## Conclusion

Overall, this algorithm for the treatment of the equation of particle motion is precise and robust enough to allow simulations even in complex cases (here laminar-turbulent transition).

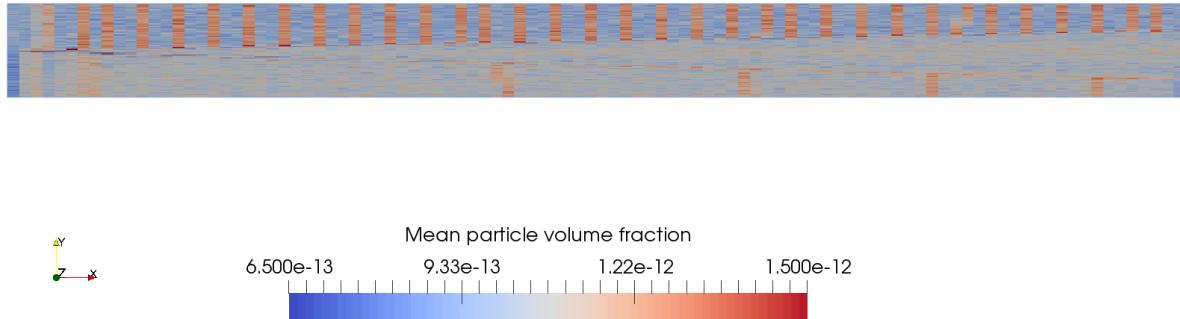


Figure 5.27: Particle volume fraction in the plane mixing layer with a correction for  $T_L$ .

**Perspectives** The validation case highlights some limitations of the present simulations. In particular, two effects should be investigated in the future: the effect of the inlet conditions on the volume fraction, the origin of the increase in the  $k/\varepsilon$  ratio in the fluid phase. Regarding this second point, numerical tests have been performed enforcing a very small value of  $k/\varepsilon$  at the inlet but results showed the existence of non-negligible values of  $k/\varepsilon$  in the laminar region far from the inlet (after 5 m).

# List of Figures

5.1	Snapshot showing the positions of particles in a U-bend geometry with warped faces.	3
5.2	Sketch showing a particle displacement from $O$ to $D$ within a cell . . . . .	5
5.3	Sketch showing a particle displacement from $O$ to $D$ going through a warped face	5
5.4	3D view of the geometry of the computational domain, showing both the boundary conditions (a) and the mesh structure with warped faces close to the inlet (b). . .	7
5.5	Slice of the U-bend geometry showing the fluid velocity obtained from the fluid-flow simulation . . . . .	10
5.6	3D view of the U-bend two-phase flow simulation showing both a slice of the fluid velocity inside the pipe and the particles (colored with the residence time) at various elapsed time. . . . .	11
5.7	Evolution of the number of particles in the simulation (left) and the number of lost particles (right) for the three cases considered here for boundary conditions.	12
5.8	3D view of the single cell used for the verification of the numerical scheme in ideal cases. . . . .	18
5.9	3D view of the particles inside single cell (zoom) at the initial and final time step.	21
5.10	Comparison of numerical results for system 5.16 (symbols for 1 <sup>st</sup> and 2 <sup>nd</sup> order schemes) and analytical solutions (lines) in the general case. . . . .	22
5.11	Comparison of numerical results for system 5.16 (symbols for 1 <sup>st</sup> and 2 <sup>nd</sup> order schemes) and analytical solutions (lines) in the limit case I. . . . .	23
5.12	Comparison of numerical results for system 5.16 (symbols for 1 <sup>st</sup> and 2 <sup>nd</sup> order schemes) and analytical solutions (lines) in the limit case II. . . . .	24
5.13	Comparison of numerical results for system 5.16 (symbols for 1 <sup>st</sup> and 2 <sup>nd</sup> order schemes) and analytical solutions (lines) in the limit case III. . . . .	25
5.14	Sketch of a plane 2D mixing layer showing the characteristic convective velocity $U_c$ and the characteristic velocity difference $U_s$ , the characteristic width of the flow $\delta(x)$ and reference lateral position $\bar{y}(x)$ . . . . .	26
5.15	2D view of the geometry of the computational domain for the mixing layer simulation. . . . .	28

5.16 Snapshot of the fluid velocity obtained from the fluid-flow simulation inside the 2D rectangular duct showing the formation and growth of the mixing layer (vertical lines are used to extract and plot data on specific regions) . . . . .	31
5.17 Scaled mean velocity profile in a plane mixing layer: comparison between experimental data (symbols) and numerical results sampled at various distances from the inlet (lines) . . . . .	31
5.18 Scaled Reynolds stresses in a plane mixing layer: comparison between experimental data (symbols) and numerical results sampled at various distances from the inlet (lines) . . . . .	32
5.19 Particle volume fraction in the plane mixing layer sampled along the middle of the duct and at various distances from the inlet. . . . .	33
5.20 Particle volume fraction in the plane mixing layer showing the depletion region. .	33
5.21 Scaled mean velocity and Reynolds stresses profiles in a plane mixing layer: comparison between numerical results for the fluid phase (lines) and for the dispersed phase (symbols) sampled at various distances from the inlet. . . . .	34
5.22 Particle volume fraction in the plane mixing layer sampled along the middle of the duct and at various distances from the inlet with additional corrections for the incompressibility condition. . . . .	35
5.23 Particle volume fraction in the plane mixing layer at various distances from the inlet: comparison between the standard version and the one with the correction for $\langle U_s \rangle$ . . . . .	35
5.24 Particle volume fraction in the plane mixing layer with additional corrections for the incompressibility condition. . . . .	35
5.25 Value of the fluid Lagrangian integral timescale $T_L$ in the plane mixing layer at various distances from the inlet. . . . .	36
5.26 Scaled mean velocity and Reynolds stresses profiles in a plane mixing layer with a correction for $T_L$ : comparison between numerical results for the fluid phase (lines) and for the dispersed phase (symbols) sampled at various distances from the inlet. .	36
5.27 Particle volume fraction in the plane mixing layer with a correction for $T_L$ . . . . .	37

# List of Tables

5.1	Fluid properties used in the simulation: inlet velocity $U_{inlet}$ , Reynolds number $Re$ , dynamic viscosity $\mu_f$ and density $\rho_f$ . . . . .	6
5.2	Properties resulting from the parameters used in the simulation (Cournat, Fourier numbers). . . . .	8
5.3	Analytical solutions to system (5.4) for time-independent coefficients. . . . .	15
5.4	Weak first-order scheme (Euler scheme) . . . . .	16
5.5	Weak second-order scheme . . . . .	17
5.6	Fluid properties used in the simulation: inlet velocity $U_{inlet}$ , Reynolds number $Re$ , dynamic viscosity $\mu_f$ and density $\rho_f$ . . . . .	19
5.7	Values of the fixed parameters in the various cases studied. General case: $\Delta t \ll T_L, \tau_p$ . Limit case I: $\tau_p \ll \Delta t \ll T_L$ . Limit case II: $T_L \ll \Delta t \ll \tau_p$ . Limit case III: $\tau_p, T_L \ll \Delta t$ . . . . .	21
5.8	Fluid properties used in the simulation: inlet velocity $U_{inlet}$ , dynamic viscosity $\mu_f$ and density $\rho_f$ . . . . .	28
5.9	Properties resulting from the parameters used in the simulation (Cournat, Fourier numbers). . . . .	29

# Bibliography

- [1] C. Henry, K. Norrfors, M. Olejnik, M. Bouby, J. Luetzenkirchen, S. Wold, and J.-P. Minier. A refined algorithm to simulate latex colloid agglomeration at high ionic strength. *Adsorption*, 22:503–515, 2016.
- [2] E. Peirano, S. Chibbaro, J. Pozorski, and J.-P. Minier. Mean-field/PDF numerical approach for polydispersed turbulent two-phase flows. *Progress in Energy and Combustion Science*, 32(3):315–371, 2006.
- [3] J.-P. Minier, E. Peirano, and S. Chibbaro. Weak first and second order numerical schemes for stochastic differential equations appearing in lagrangian two-phase flow modelling. *Monte Carlo Methods and Applications*, 9(2):93–133, 2003.
- [4] S. B. Pope. *Turbulent flows*. Cambridge University Press, 2000.
- [5] F.H. Champagne, Y.H. Pao, and I.J. Wygnanski. On the two-dimensional mixing region. *Journal of Fluid Mechanics*, 74:209–250, 1976.