Advection-diffusion with sharp interfaces, coupled with a fluid

Jonathan LEMUS

Department of Computer Science, University of Geneva, 1227 Carouge, Switzerland

Department of Earth Sciences, University of Geneva, 1205 Geneva, Switzerland

Palabos Summer school 2021

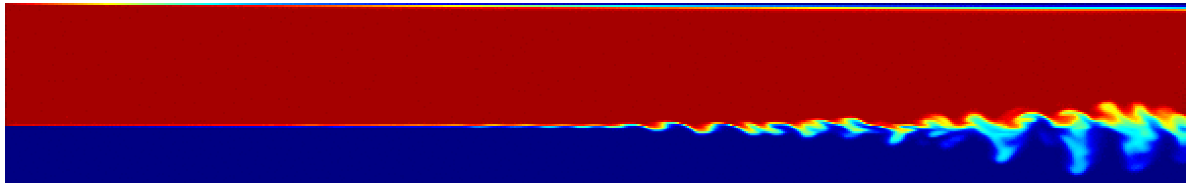
Tuesday, June 8th

Introduction

- Lattice Boltzmann Method (+ HPC): significant achievements in CFD → modelling complex flows (multiphase fluids, thermal and buoyancy effects…)

- Advection-diffusion (AD) = common problem to simulate transport of quantities such as temperature, concentration of species…

Rayleigh-Bénard instability
N=100; Ra=1E+08; Pr=1



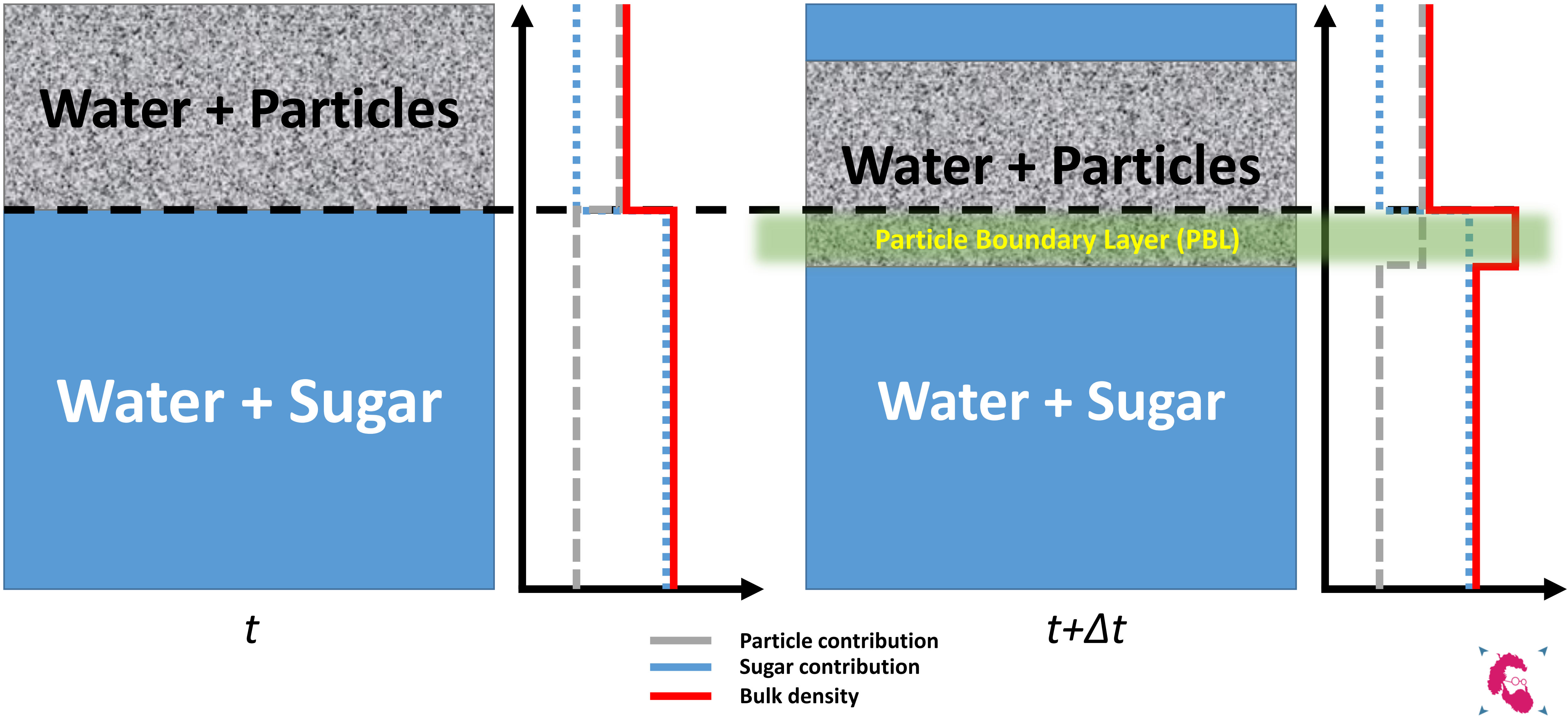
Lateral transport of volcanic ash in air

Introduction

- Advection-dominated problems = challenge for mesh-based schemes
- An example : transport of particles (modelled by a continuum field under some assumptions) with very low diffusivity.
- Interesting in order to avoid simulating individually very large number of particles
- If we add on top some sharp interfaces ➔ problems related to numerical diffusion, dispersion…

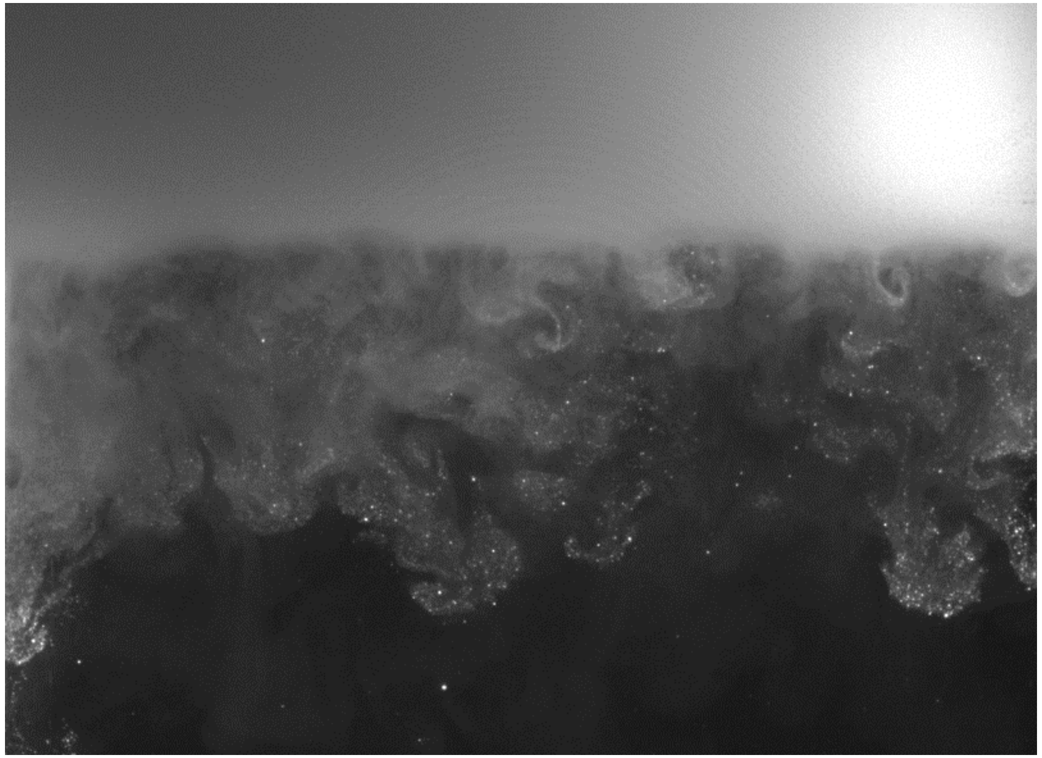
Coupled system

- <u>Case study</u>: A settling-driven Rayleigh-Taylor problem



Coupled system

- Unstable PBL collapses → downward moving fingers



Laboratory experiments on settling-driven
Raleigh-Taylor (Credits: Allan FRIES)

Coupled system - Governing equations

- How to model this ?

➔Three-way coupling between fluid, sugar and particles.

  - Fluid (momentum): $\frac{\partial \vec{u}_f}{\partial t} + (\vec{u}_f.\vec{\nabla})\vec{u}_f = -\frac{1}{\rho_0}\vec{\nabla}p + \nu\nabla^2\vec{u}_f + \left[\left(\frac{\rho_p-\rho_0}{\rho_0}\right)\emptyset + \left(\frac{\rho(S)}{\rho_0} - 1\right)(1-\emptyset)\right]\vec{g}$

  - Particles : $\frac{\partial \emptyset}{\partial t} + \vec{\nabla}.\left[\emptyset\left(\vec{u}_f - V_s(S)\vec{e_z}\right)\right] = D_c\nabla^2\emptyset$

  - Sugar : $\frac{\partial \rho}{\partial t} + \vec{u}_f.\overrightarrow{\nabla\rho} = D_S\nabla^2\rho$ *(assuming $\rho = \rho_0(1+\alpha S)$)*

- Assumptions to model particles as a continuum:
  - Small particles and large number
  - Drag force in equilibrium with gravitational force (Boussinesq)

- Particles have very small diffusivity ➔ Convection-dominated problem !

Coupled system - Governing equations

How to solve the advection-diffusion equations ?

Numerical schemes for Advection-Diffusion

1. <u>LBM for the Advection-Diffusion</u>

- Similar to Navier-Stokes equations (which can be seen as the AD of the momentum density vector $\rho\vec{u}$)

- For instance the BGK model, the relaxation time in the LB equation is given by:

$$\tau = \frac{D}{c_s^2} + \frac{\delta t}{2}$$

- All dynamics related to AD using LBM are in `palabos\src\complexDynamics\advectionDiffusionDynamics.h`

Numerical schemes for Advection-Diffusion

2. Upwind Finite-Difference 1st order:

- Adaptive scheme which takes in account the velocity direction in order to discretize (by a Taylor expansion) the convective term

$$u\frac{\partial a}{\partial x}$$

  - If $u > 0$ we use the backward difference

$$\left(\frac{\partial a}{\partial x}\right)_B \approx \frac{a_i^n - a_{i-1}^n}{\delta x}$$

  - If $u < 0$ we use the forward difference

$$\left(\frac{\partial a}{\partial x}\right)_F \approx \frac{a_{i+1}^n - a_i^n}{\delta x}$$

Numerical schemes for Advection-Diffusion

3. <u>Upwind Finite-Difference 2nd order:</u>

- If $u > 0$ we use the backward difference

$$\left(\frac{\partial a}{\partial x}\right)_B \approx \frac{3a_i^n - 4a_{i-1}^n + a_{i-2}^n}{2\delta x}$$

- If $u < 0$ we use the forward difference

$$\left(\frac{\partial a}{\partial x}\right)_F \approx \frac{-a_{i+2}^n + 4a_{i+1}^n - 3a_i^n}{2\delta x}$$

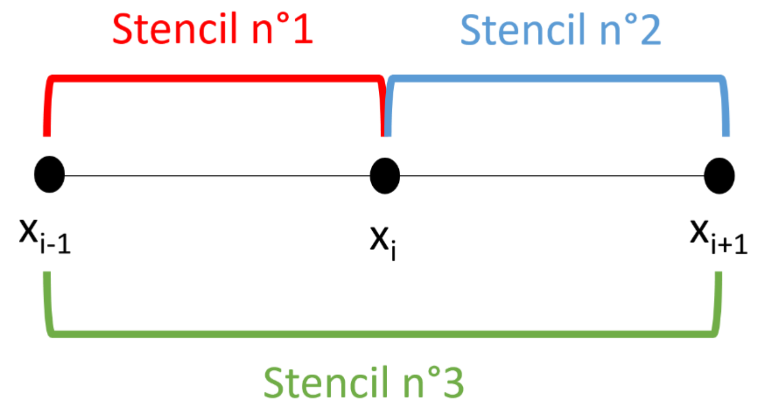
- Note that here we need 2 neighbours, which is important when initializing the scalar field associated (envelope)

Numerical schemes for Advection-Diffusion

4. WENO 3rd order

- Allows to solve hyperbolic partial differential equations of the form $\frac{\partial \emptyset}{\partial t} + \frac{\partial h}{\partial x} = 0$ where, in our case, the flux $h(\emptyset) = u\emptyset$ (Jiang and Shu, 1996; Liu et al. 1994)
- Ensures the suppression of spurious oscillations around discontinuities. Third order accurate in smooth regions and second order near discontinuities.
- Principle: build a convex combination of interpolants for the flux at given points of the domain, using two adjacent stencils of two points each
- Discretisation of the convective term:

$$\frac{\partial h}{\partial x} = \frac{h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}}}{\delta x}$$



- $h_{i+\frac{1}{2}}$ is the flux at the half node location $x_{i+\frac{1}{2}}$.
- 3 ways to approximate $h_{i+\frac{1}{2}}$:
  - $h^{(1)}_{i+\frac{1}{2}}$: interpolation with polynomials of degree 1 in Stencil 1 (second order accurate)
  - $h^{(2)}_{i+\frac{1}{2}}$: interpolation with polynomials of degree 1 in Stencil 2 (second order accurate)
  - $h^{(3)}_{i+\frac{1}{2}}$: interpolation with polynomials of degree 2 in Stencil 3 (third order accurate)

4. WENO 3rd order

- If h is smooth in the Stencil 3, we can write $h^{(3)}_{i+\frac{1}{2}}$ as a linear combination of the two second order approximations:

$$h^{(3)}_{i+\frac{1}{2}} = \gamma_1 h^{(1)}_{i+\frac{1}{2}} + \gamma_2 h^{(2)}_{i+\frac{1}{2}}$$

where $\gamma_1 = 3/4$ and $\gamma_2 = 1/4$

- What if there is a discontinuity in one of the stencils ?

The linear coefficients are replaced by « smoothness indicators » and we have:

$$h_{i+\frac{1}{2}} = \omega_1 h^{(1)}_{i+\frac{1}{2}} + \omega_2 h^{(2)}_{i+\frac{1}{2}}$$

If h is smooth in all the stencils, $\omega_i \to \gamma_i$ and ensures a third-order accuracy. Conversely, the presence of any discontinuity in the *i-th* stencil means $\omega_i \to 0$, decreasing the accuracy to second-order.

Numerical schemes for Advection-Diffusion

What about the time discretization ?

- For the 1st (and higher) order finite difference, we use the Euler explicit discretization for the time.

- For the WENO scheme (3rd order), to ensure a Total Variation Diminishing (TVD) property, we perform a 3rd order Runge-Kutta procedure for the time discretization.

Coupling in Palabos

- Let’s model the fluid motion using the LBM. Examples of dynamics taking in account a force term available in Palabos:
  - `NaiveExternalForceBGKdynamics`, `GuoExternalForceBGKdynamics`, `GuoExternalForceConsistentSmagorinskyCompleteRegularizedBGKdynamics`…

- Two “levels” of coupling:
  - Introduce the fluid velocity field in the convective term of the advection-diffusion
  - Implement the buoyant force term (which depends on the particle volume fraction and the sugar concentration)

Coupling in Palabos (Velocity)

```cpp
template< typename T, template<typename U1> class FluidDescriptor, template<typename U2> class VolfracDescriptor>
void VelocityCouplingProcessor2D<T,FluidDescriptor,VolfracDescriptor>::process (
        Box2D domain,
        BlockLattice2D<T,FluidDescriptor>& fluid,
        BlockLattice2D<T,VolfracDescriptor>& VolFrac )
{
    enum {velOffset   = VolfracDescriptor<T>::ExternalField::velocityBeginsAt,};
    Dot2D offset = computeRelativeDisplacement(fluid, VolFrac);
    for (plint iX=domain.x0; iX<=domain.x1; ++iX){
        for (plint iY=domain.y0; iY<=domain.y1; ++iY){
            T *u = VolFrac.get(iX+offset.x,iY+offset.y).getExternal(velOffset);
            Array<T,FluidDescriptor<T>::d> vel, vel_tot;
            fluid.get(iX,iY).computeVelocity(vel);
            vel_tot = vel + vsed;
            vel_tot.to_cArray(u);
        }
    }
}
```

Coupling in Palabos (Force term)

```
template< typename T, template<typename U> class FluidDescriptor >
void ForceTermProcessor2D<T,FluidDescriptor>::processGenericBlocks (
        Box2D domain, std::vector<AtomicBlock2D*> fields )
{
    typedef FluidDescriptor<T> D;
    enum {
        forceOffset = FluidDescriptor<T>::ExternalField::forceBeginsAt
    };
    PLB_PRECONDITION(fields.size()==3);
    BlockLattice2D<T, FluidDescriptor>* fluid = dynamic_cast<BlockLattice2D<T, FluidDescriptor>*>(fields[0]);
    ScalarField2D<T>* volfracfield = dynamic_cast<ScalarField2D<T>*>(fields[1]);
    ScalarField2D<T>* densityfield = dynamic_cast<ScalarField2D<T>*>(fields[2]);
    Dot2D offset1 = computeRelativeDisplacement(*fluid, *volfracfield);
    Dot2D offset2 = computeRelativeDisplacement(*fluid, *densityfield);

    for (plint iX=domain.x0; iX<=domain.x1; ++iX)
    {
        for (plint iY=domain.y0; iY<=domain.y1; ++iY)
        {
            T *force = fluid->get(iX,iY).getExternal(forceOffset);
            for (pluint iD = 0; iD < D::d; ++iD)
            {
                force[iD] = /* Implement the force term here */;
            }
        }
    }
}
```

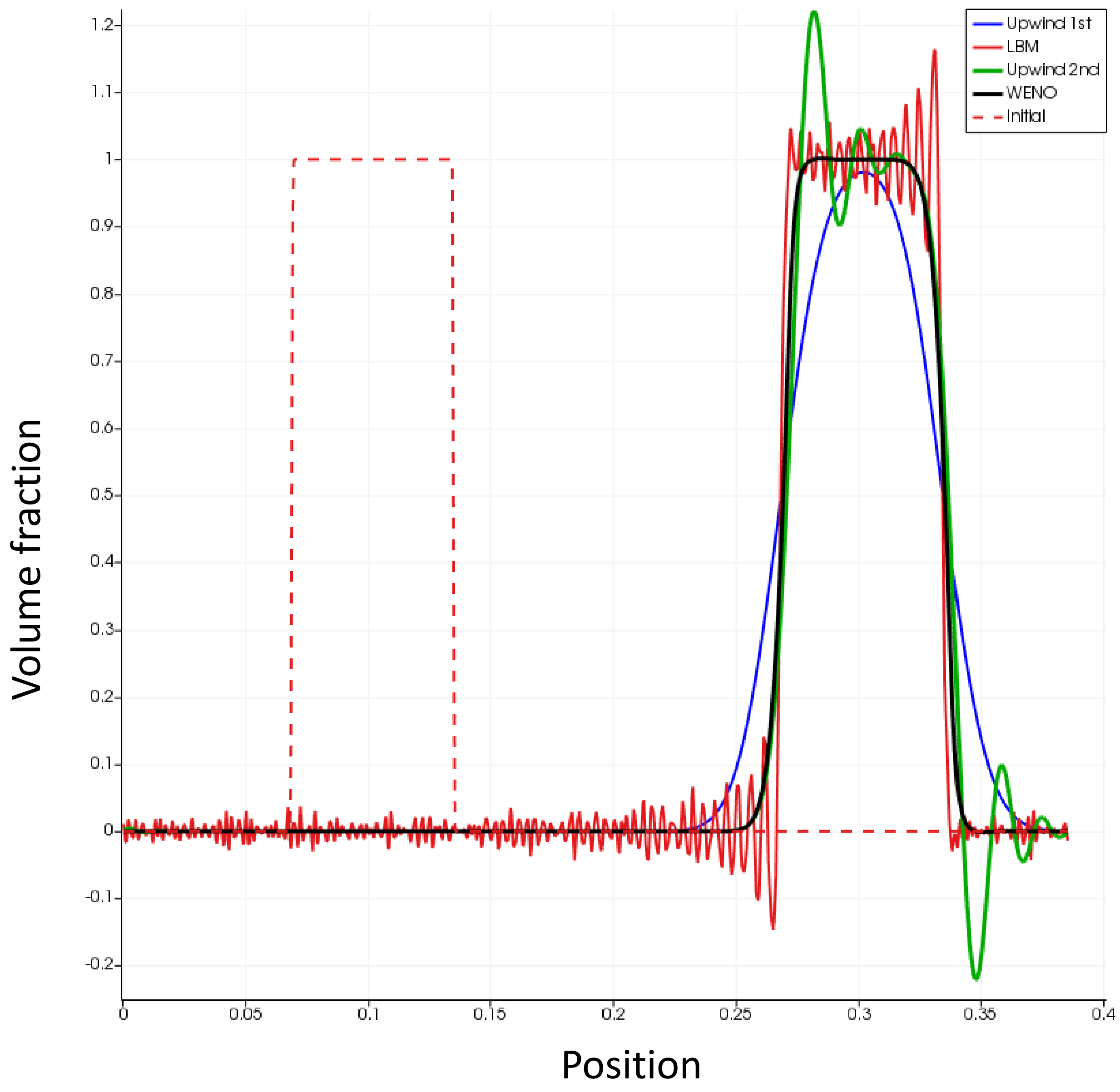
Application to problems with sharp interfaces

- Reminder for the LBM-BGK: $\tau = \frac{D}{c_s^2} + \frac{\delta t}{2}$

For very low diffusivity D, the scheme tends toward the stability limit and some spurious oscillations appear in the domain.

- Godunov theorem: *"Linear numerical schemes for solving partial differential equations (PDE's), having the property of not generating new extrema (monotone scheme), can be at most first-order accurate"*
- The upwind 1st order = stable but significant numerical diffusion (cf. Taylor expansion, the error term is $\sim u \frac{\delta x}{2} \frac{\partial^2}{\partial x^2}$)
- The second order: less numerical diffusion but introduces dispersion around interfaces (Godunov theorem !).
- <u>The 3rd order WENO is stable, and provides reduced numerical diffusion</u> (second-order accurate next to discontinuities). Possible to get higher order (e.g. WENO 5th order) but also higher computational cost.

Application to problems with sharp interfaces

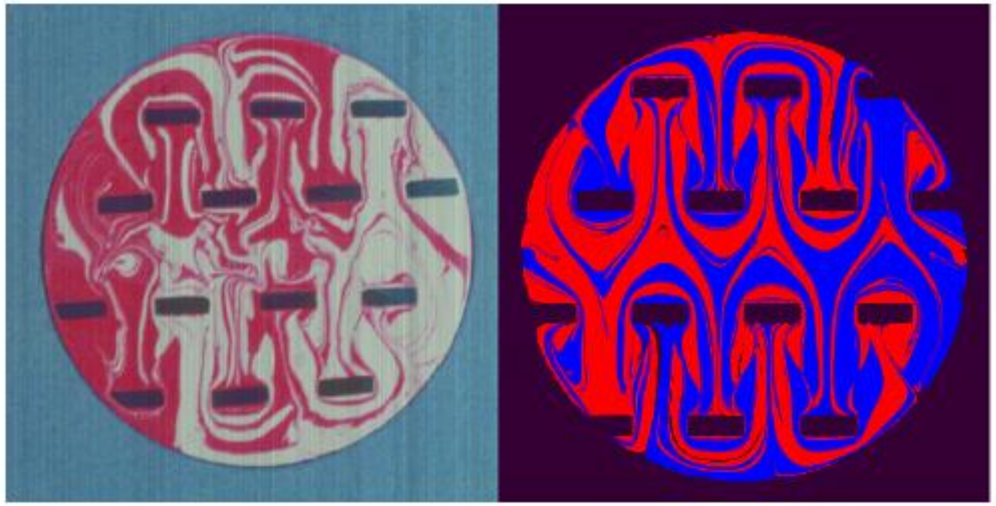


Conclusions

- Advection-diffusion possible using a full LBM scheme
- Capability to get hybrid models by coupling the LBM with finite-difference schemes ➔ more stable (at least when D<<1)
- The upwind schemes provide a stables model and are accurate for non convection-dominated problems and without sharp interfaces.
- The WENO procedure is interesting in order to model those problems, keeping the ease of implementation on uniform meshes.

Conclusions

- Particle-based approach: we don't solve any advection-diffusion equation.
- Injection of particle tracers in very large number in order to accurately follow the interface between different inviscid phases



J. Latt, D. Kontaxakis, L. Chatagny, F. Muggli, and B. Chopard, **Hybrid Lattice Boltzmann Method for the Simulation of Blending Process in Static Mixers**, *International Journal of Modern Physics C*, vol. 24, no. 12, p. 1340009, 2013

Exercises

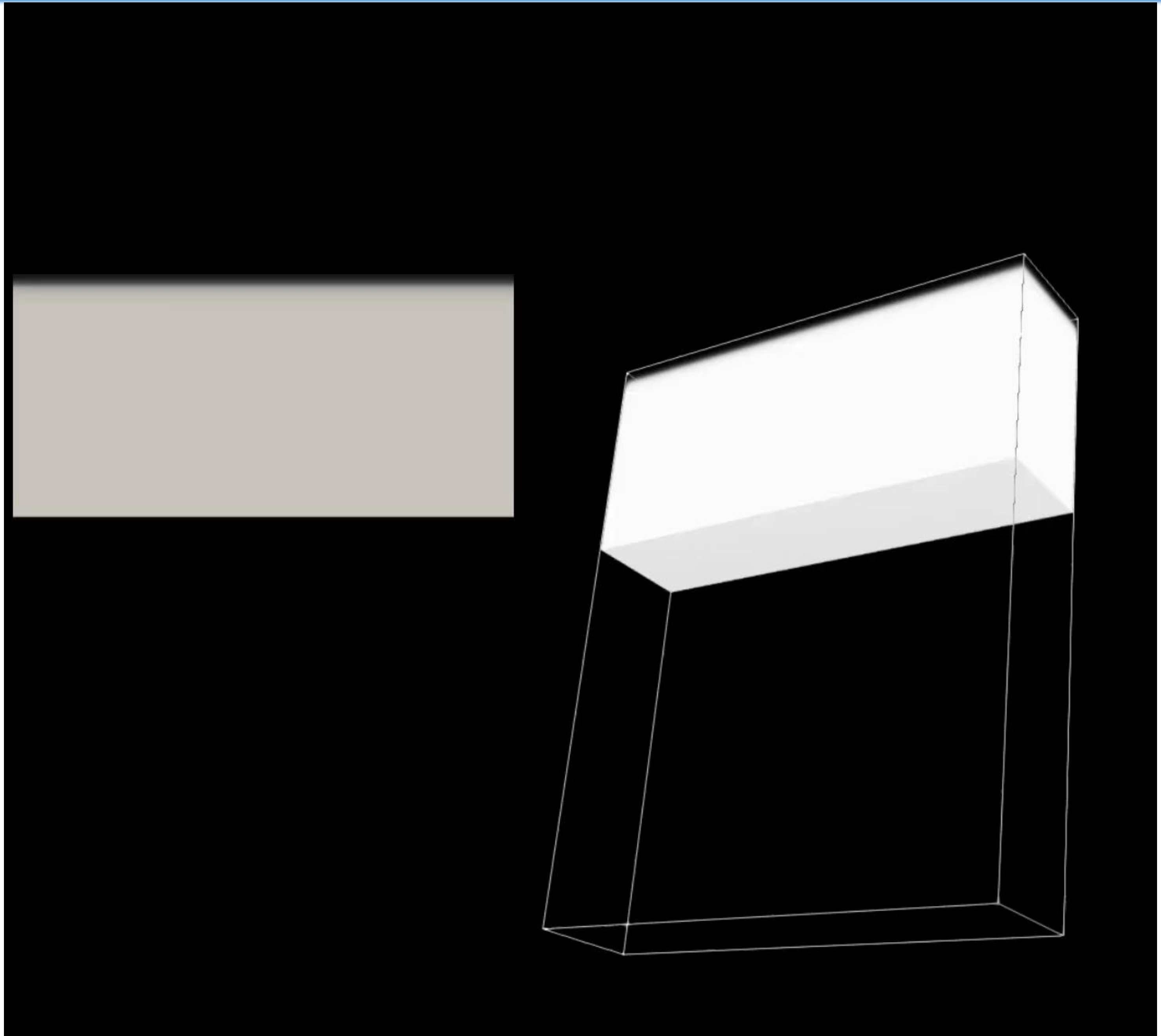
- Exercise 1:
  - Familiarize with the different Finite-difference schemes and see how they can be implemented in *Palabos.*
  - Compare them in order to highlights the main differences.
  - Check how the velocity coupling is implemented.

- Exercise 2:
  - More complex problem simulating our case study in 2D
  - Check the force term coupling and try to implement it:

$$\vec{F} = \left[\left(\frac{\rho_p - \rho_0}{\rho_0}\right)\emptyset + \left(\frac{\rho}{\rho_0} - 1\right)(1 - \emptyset)\right]\vec{g}$$

  - Run the simulation in order to observe the growing instability at the interface

Application to problems with sharp interfaces



THANK YOU !
QUESTIONS ?